https://doi.org/10.2352/ISSN.2470-1173.2020.7.ISS-145

Event threshold modulation in dynamic vision spiking imagers for data throughput reduction

Luis Cubero, Arnaud Peizerat, Dominique Morche, Gilles Sicard; CEA-LETI; Grenoble, France

Abstract

Dynamic vision sensors are growing in popularity for Computer Vision and moving scenes: its output is a stream of events reflecting temporal lighting changes, instead of absolute values. One of its advantages is fast detection of events, which are asynchronously read as spikes. However, high event data throughput implies an increasing workload for the read-out. That can lead to data loss or to prohibitively large power consumption for constrained devices. This work presents a scheme to reduce data throughput by using near pixel pre-processing: less events codifying temporal change and intensity slope magnitude are generated. Our simulated example depicts a data throughput reduction down to 14 %, in the case of the most aggressive version of our approach.

Introduction

Artificial intelligence and Computer Vision algorithms related to dynamic scenes are key enablers for autonomous driving or motion analysis, among others. Most classic approaches are frame based (FB) only, and dynamic information is recovered by processing from at least 2 intensity images. However, there is a rising interest in non-standard solutions for overcoming several limitations of this approach. For example, [1] mentioned that optical flow estimation by using all pixels, even the non-changing ones (or redundant), from subsequent frames increases latency, thus restricting control of fast movements. Indeed, non-standard imagers optimized to catch motion have been proposed. For example, Dynamic Vision Sensors (DVS) [2] [3] [4] [5] [6] have outputs that reflect temporal relative light intensity changes, namely events, and only pixels that detected an event are read. Voltage V_{diff} in figure 1 reflects this temporal difference [4], and is compared with a global reference: the contrast event threshold (CT) [4]. CT is proportional to the reference voltages V_{OFF}, and V_{ON} in figure 1 multiplied by C_2/C_1 [2]. If $|\, V_{diff} \,| > CT$, then a spike is generated and read asynchronously [2]. Also, in order to calculate V_{diff} along time, a reference value is sampled at capacitor C_2 in fig. 1 and reset locally each time a spike is generated [2]. In this work, we represent the influence of this reference value on V_{diff} as I_{ref} (see fig. 1). In general, a reading scheme called the Address Event Representation (AER) is used [2]. Being the event latency very small (around 15 μs [3]), the DVS is interesting, for instance, for simultaneous localization and mapping (SLAM) [7], optical flow estimation [1] [8], proximity detection for mobile devices [9], and region proposals for object localization with Spiking Neural Networks [10].

However, DVS versions have limited maximum (peak) data throughput (DT), reported in mega events per second or Me/s [3] [4] [5] [6]. For example: 50 Me/s at 240x180 resolution [5], or 300 Me/s at 640x480 resolution [6]. Even if the peak DT is enough for a realistic scene activity, power consumption increases with it: power rises by a factor of 2,8 [5] or 1,85 [6] from low to high DT. Power efficiency has been addressed by [11]: the asynchronous reading scheme is changed for a framed based one, obtaining an *equivalent* peak DT of 180 Me/s. Logic (for each group of 4 pixels) and memory (per pixel) units are also included, to reduce pixels read-out activity by means of redundancies suppression. Nevertheless, the equivalent peak DT is still limited, and the time resolution (equivalent to the event latency) is dictated by the event-frames per second. For example, for their reported power consumption of 250 μW for a 132x104 resolution, related to 0.1 Me/s (at 1k event-frames per second), events are not longer read with a latency in the order of several or tens of μs, but at each ms. That limits the time resolution at which fast motions can be processed.

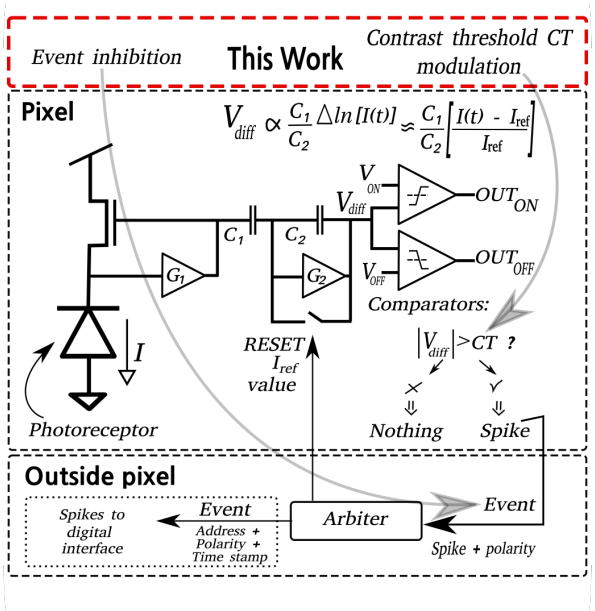


Figure 1: DVS system representation based on [3], and the red (upper) rectangle indicates our contribution.

This work presents a technique that allows reducing DT without reducing the time resolution. This technique generates less asynchronous events and codifies both temporal change and intensity slope magnitude. The main objective is to diminish the read-out and digital processing workloads. For that purpose, a CT modulation, and an inhibition system (controlled by a DISABLE / ENABLE signal) are proposed. Pixel schematic modifications are presented and explained, where the pixel complexity (which

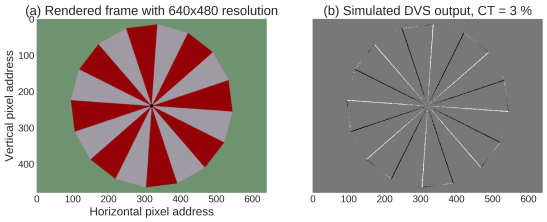


Figure 2: (a) Rendered rotating polygon, and (b) Simulated DVS events during 15 μs. Positive-slope events are in white, and negative-slope events are in black.

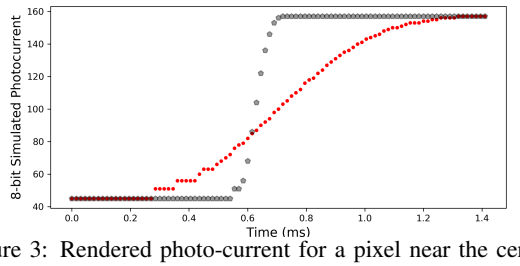


Figure 3: Rendered photo-current for a pixel near the center of the rotating polygon in fig. 2 (red dots), or far from it (gray pentagons).

is related to the imager fill factor) is kept as compact as possible.

This paper is organized as follows: firstly, we introduce our simulation environment. Then, we explain how to generate sets of events accordingly to their CT. After, we illustrate the DT reduction, and we present the pixel circuit modifications that are needed for our proposed modulation and inhibition scheme. Finally, conclusion are drawn.

Our simulation environment

One simple rotating polygon (counter-clockwise) scene was rendered with an open source tool for 3D graphics modeling: Blender [12]. The scene was inspired from the work of [8] (they used a disk instead of a polygon). 1000 frames were obtained, corresponding to a rotation angle of 37 degrees. For instance, considering a time resolution dictated by an event read latency of 15 μs, our scene represents an angular rotation of $\approx 43\,rad/s$ (which is approximately the angular speed of a car wheel of $0.5\,m$ diameter, when the car is moving along a straight line at $\approx 77.5\,km/h$). The cycles rendering mode was used, which takes into account light reflections and propagation [13]. This sort of rendering introduced noise due to the finite amount of light rays used [14]. Then, an DVS simulator developed in our previous work [15] was set with a CT of 3 %, which is realistic accordingly to the literature [4]. The DVS behavior was approximated by using the rendered frames as input to our simulator, and our simulation output was the events as addresses (x, y) plus their time stamps. Those events can be accumulated during a certain time in order to be represented in an image, as in fig. 2.(b).

In fig. 3, for two different pixels, photo-current intensity is plot for the time interval during which a color edge is sensed. Those pixels were in the same vertical sensor line, but one was closer to the center, whereas the other closer to the perimeter. A point along a color edge moves faster as going further away from the disk center. That is translated into a higher slope in the light intensity curve.

CT modulation and read-out scheme

The idea is to generate less spikes in the case of high intensity slopes. For lower intensity slopes, spikes generation related to small changes are kept without modifications. Our modulation is depicted in figure 4. We call an imager compatible with this modulation the mDVS (or modulated DVS).

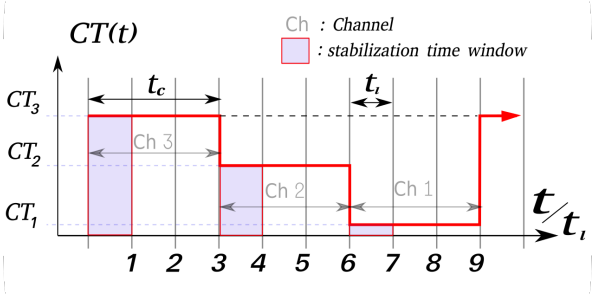


Figure 4: Timing diagram for our proposed CT modulation.

The time resolution is determined by the event reading latency t_l, and the idea is to change the contrast threshold CT from an upper to a lower values as shown in figure 4. In this example, t_c is the duration for which CT is constant. After a new CT is set, events within t_l are related to the stabilization period, as it is further explained in section V. After t_l is passed, arriving events can be classified in channels. For example, if 3 CT are set, that would lead to 3 channels. In our simulations t_c was set to three times t_l.

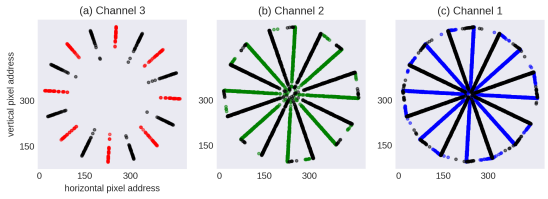


Figure 5: Example of outputs for different channels with different CTs after applying the modulation.

Obtaining 3 channels in parallel, instead of modulating CT, could be interesting as well since no modulation would be needed. Nevertheless, that would require increasing significantly the pixel complexity, and with a high impact on the Fill Factor, DT and power consumption. The result of the CT modulation is shown in figure 5: Channel 3 does not present spikes for pixels near the center. Since lighting conditions are similar along the radial color edges, a higher CT is related to higher speeds. Then, Channel 3 reports pixels around regions of radial edges that are moving faster. Channel 1 is similar to a DVS with CT = 3 %. Channel 2 reports pixels around the radial edges, and some along the perimeter.

Stabilization time

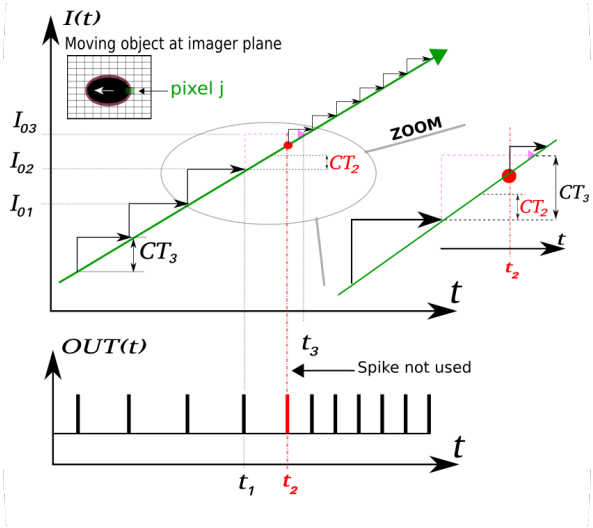


Figure 6: Ideal light intensity curve reflecting motion of an object from right to left.

Stabilization time windows (after each CT changing) correspond to short time intervals into which pixels events are not used. Figure 6 represents an ideal photo-current curve in a DVS pixel with a relatively high CT = CT_3, and an object moving from right to left along the image plane. I_{0i} represents the reference value (I_{ref}) set after a spike. At time t_1, the amount of changing at pixel j reaches CT_3 and spikes. One can observe that, for a relatively large CT, pixels will spike less frequently, and will hold the light intensity difference during more time. In figure 6, at time t_2, CT_3 is changed to CT_2, where $CT_2 < CT_3$, causing pixel j to spike exactly at time t_2. This spike is can be potentially wrongly classified into channel 2. Then, events coming right after an abrupt CT changing are not used; they might no be useless but their potential application is lead to further contributions. This amount of time (or stabilization window) in which events are not used corresponds to one t_l after a CT changing.

Data throughput

From fig. 4 one can observe that, for the set parameters, an equivalent time of $9t_l$ is needed to complete a CT modulation cycle. It is important to notice that events are still being read asynchronously, and thus the modulation does not imply a FB read-out. During the CT modulation, events coming from a channel with higher CT will reflect higher relative temporal contrasts (i.e. photocurrent slopes). The modulation in mDVS causes DT reduction as shown in table 1. In fact, fast changing pixels spike less in average, and slow changing pixels spike only during the low CT interval. Moreover, this kind of read-out allows assigning a channel to each event, reflecting which pixels are changing with faster slopes without the need of further processing. The technique proposed by [11] has the limitation of time resolution being diminished accordingly to the frame rate.

One can also observe that when a spike from the channel 3 is generated, the amount of intensity change needed is approximately equivalent to generating several spikes from channel 2, and even more from channel 1. If one pixel spikes during the channel 3 interval, then this pixel can be inhibited since a high slope event has already been obtained. When a pixel is inhibited it can no longer generate spikes till this inhibition is removed. In a standard DVS, slope magnitude information could only be approximated by counting spikes from pixels and adding them during a time interval. Moreover, such inhibition reduces further DT, which has already been reduced compared to a standard DVS due to CT modulation. During a period of t_l after a CT changing, pixels addresses and time stamps are not used (due to the stabilization time window). After t_l, there are two modes for this inhibition: the SE-ch mDVS inhibits pixels (independently from other pixels) from spiking once they have spiked during a channel. That is, a DISABLE signal is sent asynchronously to a spiking pixel when it spikes, thus starting the inhibition. Then, an ENABLE signal is sent synchronously to all pixels after one t_l from each CT changing for stopping all inhibitions. The SE-cy mDVS inhibits pixels once they have spiked during a CT modulation period (i.e. the synchronous and global ENABLE signal is sent only after one t_l from the start of each CT modulation period, whereas the DISABLE signal behaves as in the previous case). Results are depicted in fig. 1, reflecting a DT reduction down to below 20% in our simulations.

Table 1: Simulated data throughput (DT) for the rotating disk scene, for the standard simulated DVS and for the modified pixels. Event read latency or time resolution is $15\,\mu s$.

IMAGER TYPE	DVS	mDVS	SE-ch mDVS	**SE-cy mDVS**
DT Mevents/s	259	82	47	**36**
DT/DT_{DVS} [%]	100	32	18	**14**

The SE-cy mDVS might imply the inhibition of pixels that spiked during the stabilization period as well, which may not be the optimal case. For example, when CT changes from CT_3 to CT_2 in fig. 4, some pixels would intermediately generate spikes and would be inhibited, but since those events happend during a stabilization period then they were not used. For the SE-ch mDVS this problem is not present since the inhibition is stopped after one t_l for each CT changing, and not after one whole modulation period. For the SE-cy mDVS, inhibition of pixels that spiked during stabilization periods may generate some data loss, representing a trade-off with the further reduced DT down to less than %15 in our simulation. Further works could explore the adaptability between the two inhibition types since they could be used without changing the electronics.

Pixel modifications

One DVS pixel schematic and our proposed modifications are shown in figure 7. In pixel block A is for coarse CT changes. Block B represents the pixel connectivity to the modulated reference voltages (outside pixel) for fine CT changes. Those voltages minimal range can be around 50 - 100 mV [2], and its maximum value is limited by the dynamic range of the comparator. Then, in order to increase CT from 3% to 50%, C_1 can be changed (block

A) so the factor C_2/C_1 is significantly increased (or decreased). Block C (inside pixel) takes care of the inhibition. The inhibition is sampled and held onto the capacitor C_{inh} as follows: when a pixel spikes, the arbiter circuit implements a handshake logic (as in a standard DVS) and sends a pixel DISABLE signal to start the inhibition. Please notice that in a standard DVS, the arbiter would normally send a RESET signal to reset the pixel's reference value. In our case, this signal (which we call DISABLE) will be sent to node e, letting the value of V_{diff} close to 0 V as long as C_{inh} is holding this low value. The pixel will not spike since V_{diff} is maintained below the comparators triggering thresholds. When the synchronous ENABLE signal (generated outside pixel) is applied, all inhibitions are turned inactive (restoring pixels functioning) as long as C_{inh} is holding this high value.

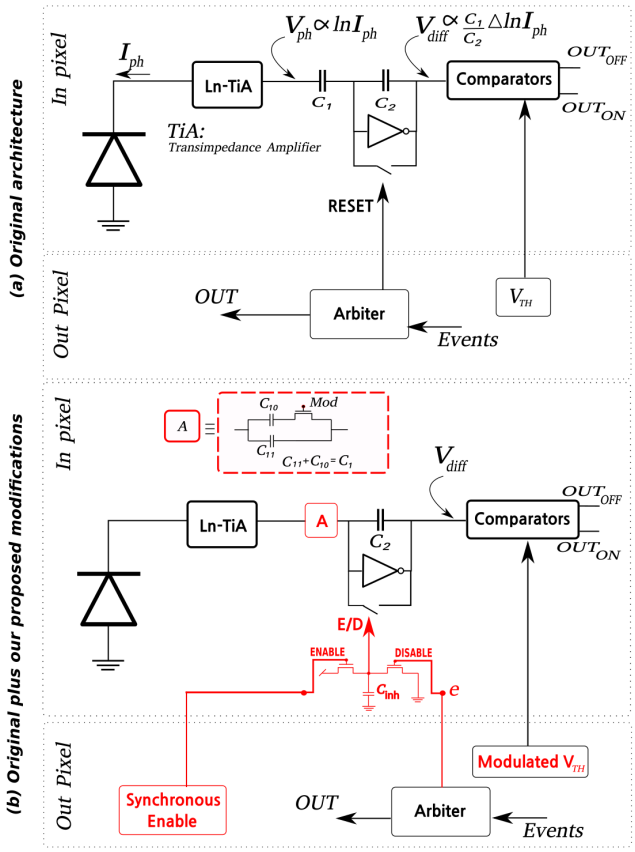


Figure 7: Original pixel schematic from [3] and our proposed modifications (in red).

Conclusion

We have proposed a modulation and inhibition scheme for dynamic vision sensors. Our approach is intended for data throughput reduction for relaxing the read-out workload, while keeping time-stamps resolution in the order of μs. Pixel modifications are presented and explained. Our simulation results depict a reduction down to 32 or 14 % in data throughput in our illustrative example, depending on the modulation variant used from our approach. In further works, our pre-processing technique could facilitate motion analysis by taking into account the channel of each event.

References

[1] B. J. Pijnacker Hordijk, K. Y. Scheper, and G. C. De Croon, "Vertical landing for micro air vehicles using event-based optical flow," *Journal of Field Robotics*, vol. 35, no. 1, pp. 69–90, 2018.

[2] C. Posch, T. Serrano-Gotarredona, B. Linares-Barranco, and T. Delbruck, "Retinomorphic event-based vision sensors: Bioinspired cameras with spiking output," *Proceedings of the IEEE*, vol. 102, no. 10, pp. 1470–1484, Oct 2014.

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

[4] T. Delbruck and R. Berner, "Temporal contrast aer pixel with 0.3%-contrast event threshold," in *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2010, pp. 2442–2445.

[5] C. Brandli, R. Berner, M. Yang, S. Liu, and T. Delbruck, "A 240 × 180 130 db 3 μs latency global shutter spatiotemporal vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2333–2341, Oct 2014.

[6] B. Son, Y. Suh, S. Kim, H. Jung, J.-S. Kim, C. Shin, K. Park, K. Lee, J. Park, J. Woo *et al.*, "4.1 a 640× 480 dynamic vision sensor with a 9μm pixel and 300meps address-event representation," in *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. IEEE, 2017, pp. 66–67.

[7] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate slam? combining events, images, and imu for robust visual slam in hdr and high-speed scenarios," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 994–1001, April 2018.

[8] F. Paredes-Vallés, K. Y. W. Scheper, and G. C. H. E. De Croon, "Unsupervised learning of a hierarchical spiking neural network for optical flow estimation: From events to global motion perception," *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[9] J. Won, H. Ryu, T. Delbruck, J. H. Lee, and J. Hu, "Proximity sensing based on a dynamic vision sensor for mobile devices," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 1, pp. 536–544, Jan 2015.

[10] J. Acharya, V. Padala, and A. Basu, "Spiking neural network based region proposal networks for neuromorphic vision sensors," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2019, pp. 1–5.

[11] C. Li, L. Longinotti, F. Corradi, and T. Delbruck, "A 132 by 104 10μm-pixel 250μw 1kefps dynamic vision sensor with pixel-parallel noise and spatial redundancy suppression," in *2019 Symposium on VLSI Circuits*, June 2019, pp. C216–C217.

[12] Blender. Internet draft. Blender. Accessed: September 26th, 2019. [Online]. Available: https://www.blender.org

[13] Blender 2.80 manual: Light paths. Internet draft. Blender. Accessed: September 19th, 2019. [Online]. Available: https://docs.blender.org/manual/en/latest/render/cycles/render_settings/light_paths.html

[14] Blender 2.80 manual: Reducing noise. Internet draft. Blender. Accessed: September 20th, 2019. [Online]. Available: https://docs.blender.org/manual/en/latest/render/cycles/optimizations/reducing_noise.html

[15] L. Cubero, A. Peizerat, D. Morche, and G. Sicard, "Smart imagers modeling and optimization framework for embedded ai applications," in *2019 15th Conference on Ph.D Research in Microelectronics and Electronics (PRIME)*, July 2019, pp. 245–248.

Author Biography

Luis Cubero received his Master diploma in Nanotechnology for Integrated Systems in 2018, from a program shared by The University of Electronics, Physics and Materials (Grenoble-INP Phelma), the Swiss Federal Institute of Technology in Lausanne (EPFL), and the Polytechnic University of Turin (Polito). He is a doctoral student at CEA-LETI, France. His research focuses in on-chip smart imaging systems design for artificial intelligence applications.

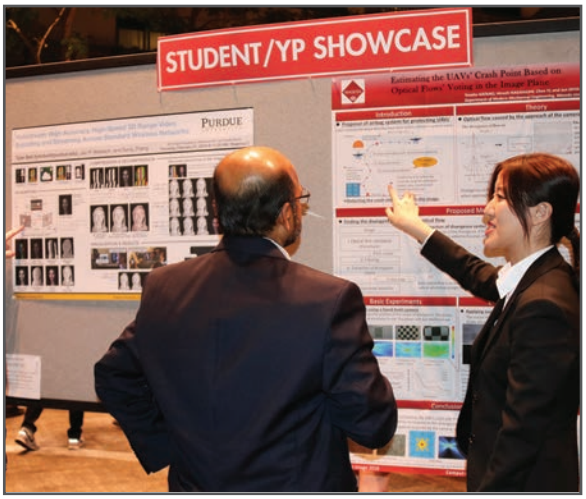
JOIN US AT THE NEXT EI!

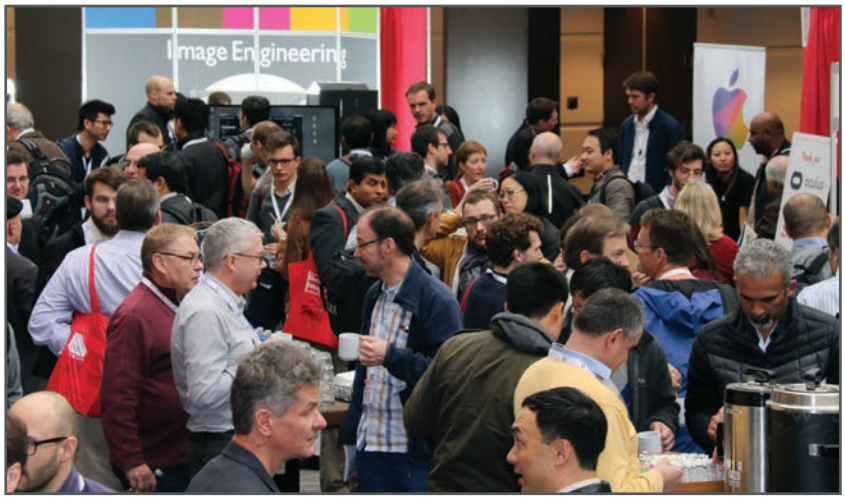
IS&T International Symposium on

Electronic Imaging

SCIENCE AND TECHNOLOGY

Imaging across applications . . . Where industry and academia meet!





• SHORT COURSES • EXHIBITS • DEMONSTRATION SESSION • PLENARY TALKS •
• INTERACTIVE PAPER SESSION • SPECIAL EVENTS • TECHNICAL SESSIONS •

www.electronicimaging.org

IS&T
imaging.org