Fun with Asynchronous Vision Sensors and Processing

Tobi Delbruck

Inst. of Neuroinformatics, University of Zurich and ETH Zurich
http://sensors.ini.uzh.ch

Abstract. This paper provides a personal perspective on our group's efforts in building event-based vision sensors, algorithms, and applications over the period 2002-2012. Some recent advances from other groups are also briefly described.

When Mahowald and Mead built the first silicon retina with asynchronous digital output around 1992 [1], conventional CMOS active pixel sensors (**APS**) were still research chips. It required the investment by industry of about a billion dollars to bring CMOS APS to high volume production. So it is no surprise that while the imager community has been consumed by the megapixel race to make nice photos, cameras that mimic more closely how the eye works have taken a long time to come to a useful form. These "silicon retinas" are much more complex at the pixel level than APS cameras and they pay the price in terms of fill factor and pixel size; machine vision cameras with capability of synchronous global electronic shutter are about 5um. Silicon retina pixels are roughly 10 times the area of a machine vision camera pixel. So why are silicon retinas still interesting? Mostly because of the high cost at the system level of processing the highly redundant data from conventional cameras, and the fixed latencies imposed by the frame intervals. High performance activity driven event-based sensors could greatly benefit applications in real time robotics, where just as in nature, latency and power are very important [2,5,9,10].

1 Being Frame Free

Like fat free milk, event-based silicon retinas can free the consumer from consumption of excess energy. To be effective, the pixels must be designed to signal significant events so that events are not redundant. For us, the story really started when we developed the first functional dynamic vision sensor (**DVS**). In the DVS, each event signifies that the log intensity has changed by some threshold amount since the last event from the pixel (Fig. 1) [2,3]. The sensor output is an asynchronous stream of pixel addresses (address-events) signifying that the brightness has increased or decreased at particular pixels. Because the event signals a log intensity change and not an absolute intensity change, it generally signifies a change of scene reflectance, which often is caused by movement of an object. This response is the key feature that makes these sensors useful for dynamic vision.

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part I, LNCS 7583, pp. 506–515, 2012.

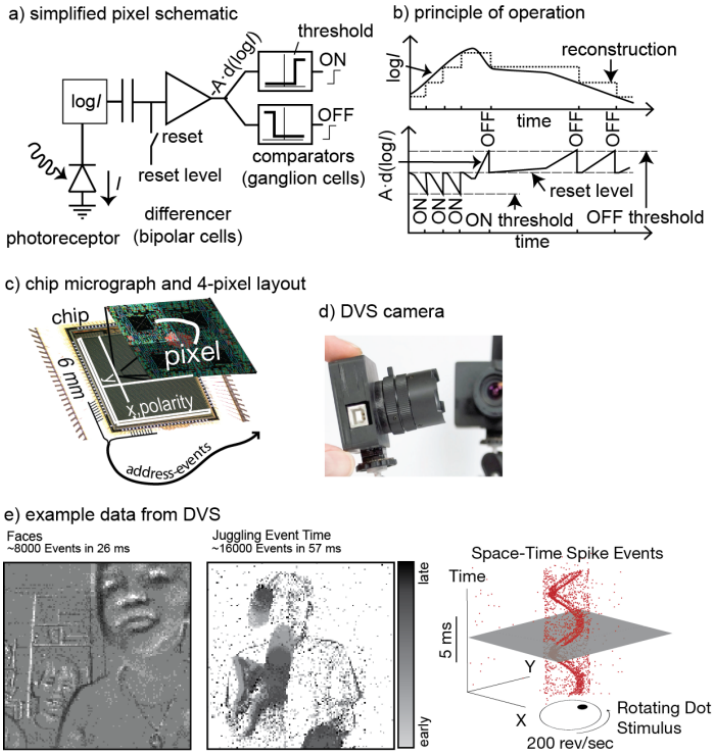

Fig. 1. The Dynamic Vision Sensor silicon retina. **(a)** The DVS pixel emulates the photoreceptor-bipolar-ganglion cell information flow. It consists of 3 parts: a logarithmic photoreceptor, a differencing amplifier (bipolar cells), and 2 decision units (ganglion cells). The pixel output consists of asynchronous ON and OFF address-events that signal scene reflectance changes. **(b)** The events are computed by the pixel as illustrated. The continuous-time photoreceptor output, which encodes intensity logarithmically, is constantly monitored for changes since the last event was emitted by the pixel. A detected change in log intensity which exceeds a threshold value results in the emission of an ON or OFF event. The threshold is typically set to about 10% contrast. Communication of the event to the periphery resets the pixel, which causes the pixel to memorize the new log intensity value. **(c)** The pixels are arranged in an array and fabricated in a standard CMOS process. Address-Event Representation (**AER**) circuits along the periphery of the chip handle the access to the shared AER bus and ensure that all events are transmitted, even if there are collisions. Colliding pixels must wait their turn for access to the AER bus. **(d)** The chips are integrated into a camera, either interfaced to a computer by USB, directly to a microcontroller, or to another neuromorphic chip via its AER interface. **(e)** Data collected from the DVS shows its characteristics: the events can be histogrammed in 2d-space over a certain time window to form an image which either displays the ON and OFF events as contrast (Faces), or as a gray scale showing the relative event time (Juggling event time), or they can be viewed in space-time to see the spatiotemporal structure (Space-Time Spike Events). Adapted from [9].

1.1 Pixel Designs

Our original DVS temporal contrast pixel design has held up remarkably well. The advantages of this design are apparent after considering a number of non-ideal effects. The pixel first of all relies on a simple continuous time logarithmic photoreceptor circuit which uses feedback to clamp the photodiode at a "virtual ground", i.e., the feedback holds the photodiode reverse bias at a fixed, small voltage, while sensing the photocurrent and outputting the result as a low impedance voltage that is logarithmic with intensity. Clamping a small reverse bias reduces dark current and thus improves dynamic range. However, we have not figured out a way to use pinned photodiodes which are standard in high performance CMOS image sensors. The voltage gain is low, only about 40mV per e-fold or 100mV per decade, but this allow representation of 7 decades of light intensity in a voltage range of less than 700mV. This means that no other gain control is necessary even in a deep sub-micron process with a supply voltage of only 1.8V. Recent designs have exposed some headroom problems that were not apparent until we encountered them in fabricated silicon. We circumvent this problem in a number of ways. The simplest solution is to use higher threshold voltage transistors in some places in the pixel. These are available in submicron processes for use in IO pads or analog circuits.

But really the key points are how the DVS achieves its sensitivity despite massive amounts of transistor mismatch. The keys are the blocking of the large DC offsets from the photoreceptor, the use of well-matched passive feedback via a capacitive divider, and the matched amplifier and comparator amplifiers. Then the gain of the amplifier is set largely by the capacitive divider ratio and not by transistor intrinsic voltage gain. The differencing amplifier and comparators are formed from 6 transistors that are all laid out in the same orientation and geometry and which are thus matched as well as it is possible to make them, if bulky common centroid layout and dummy transistors are not used.

1.2 New Retina Pixels

Bernabe Linares-Barranco and Teresa Serrano-Gotarredona at the Inst. of Microelectronics in Sevilla and Christoph Posch, now at the Vision Instititute in Paris, have been particularly creative in devising interesting retina pixels with good performance. Fig. 2 sketches the comparison discussed next.

The ATIS

Posch designed the ATIS[1] pixel with colleagues while at the Austrian Inst. of Technology [18]. This pixel consists of two sub pixels. The first sub pixel is a DVS temporal contrast pixel. Events from the DVS pixel trigger time-based intensity readings in the second sub pixel. The intensity is measured by the time is takes the photodiode voltage to integrate between two levels. The beautiful thing about this mechanism is the way it avoids both mismatch and kTC noise, by integrating not from a reset voltage to a threshold, but rather between two thresholds, which are multiplexed to a

[1] Asynchronous time-based image sensor.

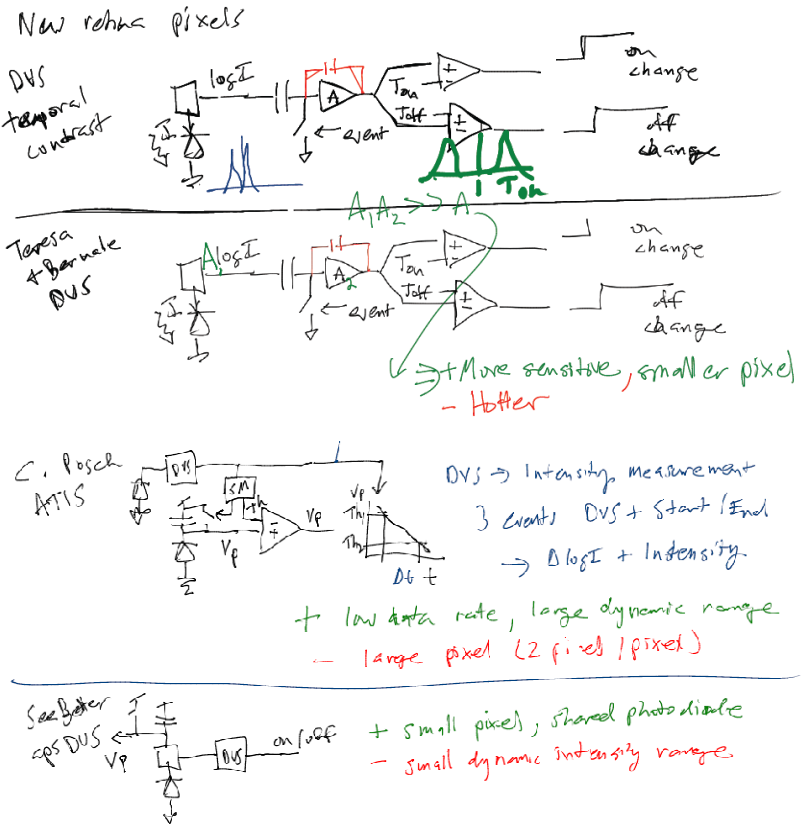

Fig. 2. Retina pixel designs that are compared in the text. I sketched this comparison at the 2012 Capo Caccia Cognitive Neuromorphic Engineering Workshop session on 5.5.12 on "Event- and Spike-based computing methods and systems."

common comparator. This way, the kTC reset level variation and the comparator offset are both suppressed [17]. The main advantage of the ATIS pixel is the event-triggered and wide dynamic range intensity readout; however the price of this is a large pixel size and small fill factor (the ATIS is effectively about twice the area of the DVS pixel and must use a separate photodiode for each measurement), and intensity capture time that can be up to several hundred ms at low intensities.

Faster and More Sensitive DVS Pixels

The latest DVS pixels from Linares-Barranco and Serrano-Gotarredona are also very interesting. They addressed the need in some applications of higher speed and sensitivity by realizing that the best improvement in performance results from adding more gain and bandwidth to the photoreceptor that precedes the differencing amplifier. They have taken two approaches to this improvement but only the first is published [19]. In their pixel, they interposed two non-inverting voltage gain amplifiers between

the logarithmic photoreceptor and the capacitive differencing amplifier. The voltage amplifiers are formed by current mirror stages using strong inversion operation with transistor geometry and operating current determining the voltage gain. This photoreceptor requires global gain control to keep the circuits in range over the entire intensity range of natural lighting. The time constant for this global gain control must be carefully chosen to provide sufficiently fast response to changes in lighting while not being so fast that it by itself generates oscillations or "gain control events". By using this circuit, they increase the gain of the photoreceptor by a factor of about 6, to result in an overall gain increase from 20 to 125. This increase allows them to set a lower nominal event threshold of about 2% contrast, compared with about 10% for our original DVS.

They also use a different feedback arrangement for the photoreceptor. Instead of supplying photocurrent from the source of an nfet with feedback to the gate of the nfet, they use the photoreceptor from Oliver Landolt [19], where the feedback photocurrent is supplied from the drain of a pfet, with feedback applied to the source of the pfet. The gate of the pfet is tied to a fixed voltage, which determines the clamped photodiode voltage. The main advantage of this circuit is the reduced Miller capacitance, which allows lower latency responses. The main disadvantages are that the photocurrent cannot be read from the drain of the transistor, and the requirement that the feedback amplifier bias must be larger than the largest photocurrent. This requirement means that bias current cannot be arbitrarily reduced to control bandwidth. However this is not a severe constraint for the high speed applications of this photoreceptor.

The apsDVS Pixel

We are trying to address some of the drawbacks of the ATIS in our newest pixel, which we call the apsDVS pixel (Fig. 3). Here "aps" stands for "active pixel sensor" and is used to describe any kind of conventional CMOS image sensor pixel with in-pixel active buffering of the integrated photodiode voltage. In our as yet unpublished apsDVS pixel, we share the same photocurrent between two complementary functions - the asynchronous detection of brightness changes and the synchronous readout of linear intensities. The cost of adding the aps readout is only 4 transistors per pixel. This pixel asynchronously emits brightness change events and we can synchronously read out the intensities by resetting and then later reading the integrated voltage.

We prototyped the first version of the apsDVS in one of our SEEBETTER chips as a 32x64 array (Fig. 3). The chip is functional but we discovered a parasitic capacitive coupling between the aps readout and the DVS circuit that generates spurious DVS events during aps readout. We currently have a corrected 240x160 design with 18.5um pixels in fabrication.

The apsDVS chip marries the advantages of simple small synchronous pixels with the low latency, wide dynamic range detection capabilities of the DVS pixels. We think the main disadvantage of the apsDVS will be the small dynamic range of the aps pixels. We hope we can take advantage of this combination in future application areas that extend on the obvious advantage of simply having a DC view of the scene in front of the sensor. In particular, we hope that we can extrapolate from the aps frames using the DVS events to complete a richer and more powerful retinal output stream than is offered by the present DVS.

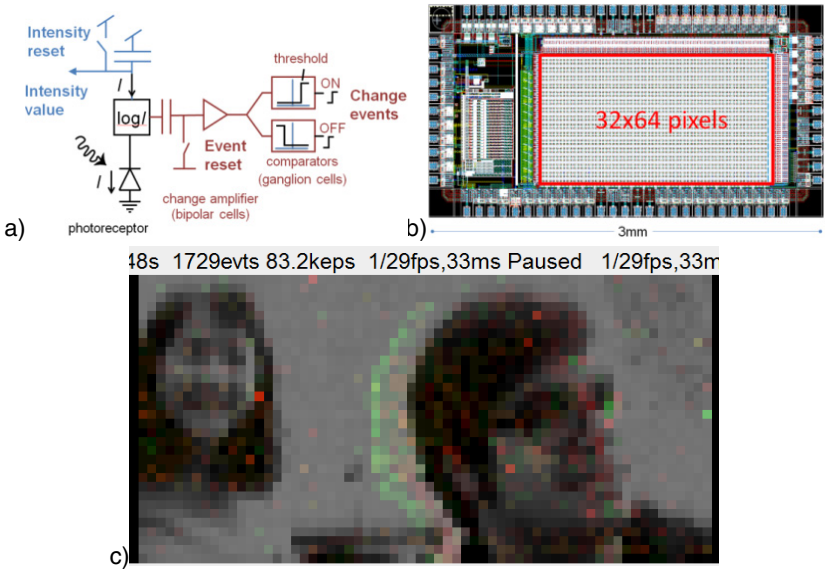

Fig. 3. The new "apsDVS" pixel. **(a)** pixel architecture; the same photocurrent provides intensity samples and temporal contrast change events; **(b)** test chip layout; **(c)** sample data over one aps frame and 18ms of DVS data; the grayscale image is from the aps pathway and the colored pixels are from events from the DVS pathway; the person is moving their head to the left and the green DVS events from the edge of his head lead the aps frame data which is from older data.

2 Usable End User Systems

We first developed the DVS in the CAVIAR project where we partnered with 4 other institutions to develop a purely hardware spike-based vision system [13]. It was our experience with the requirement in CAVIAR for a crew of 4 PhD students needed to boot and run the system that drove us strongly in the direction of software exploration of algorithms for processing sensor output. We realized that most neuromorphic labs are so heavily focused on hardware development that is rare that any device makes it off the lab bench and into the hands of potential end users. That is the main reason we put a huge effort into developing usable USB-based DVS cameras with integrated biases that are temperature and process insensitive [23].

Initially we developed event-based processing algorithms in Matlab, but we quickly realized that we needed a more structured, reusable software framework capable of multithreaded operation. These developments led in 2007 to the open source jAER software project, hosted at SourceForge [5,11]. jAER contains everything almost everyone has done with processing retina and cochlea output, and classes that encapsulate for display all of our AER chip developments and USB-based computer interfaces, along with some from other groups. As of 2012, jAER consists of more than 1000 Java classes, which makes it daunting for newcomers to understand what already exists. However the core of jAER is much smaller and our experience is that

when we have a new student, it typically requires only a few weeks for them to implement their work by sub-classing the basic event processor. But because most computer vision researchers are more accustomed to using C++ or Matlab, jAER is hard for them to grasp. In particular, what we need to develop is a simple C API that allows outsiders to easily access the raw DVS data under Windows and Linux, and following this, toolboxes that allow access to some of the algorithm outputs.

2.1 Application Areas of the DVS

Experience has shown that immediate application areas of the DVS are mostly in object tracking [4,5,6,21,22]. Here the sparse output, low latency, and form of the DVS output are ideally suited to the task of tracking moving objects. Small isolated objects like balls, cells, cars, particles in hydro or aero dynamics, etc. are easily tracked using rather simple algorithms based on updating the object models by the events; see the jAER class RectangularClusterTracker for details. The Goalie class is a complete robot implementation that tracks balls to control an arm that blocks the balls [6]. More complex objects like lines are also tracked using more sophisticated algorithms based on continuous Hough transforms. Here the algorithms become quite non-intuitive. Readers are referred to Matthew Cook's open-sourced PencilBalencer [4] and the unpublished PigTracker[2] classes for excellent examples of these algorithms. PencilBalancer is a complete implementation of a pencil balancing robot that uses a pair of DVS [4]. PigTracker extends on this idea to track an arbitrary line drawing over affine transformations including scale, rotation and skew. The goalie and pencil balancer robots run on a cheap PC with CPU load of less than 10% and latencies of about 2ms.

Surveillance and behavioral monitoring is another area of application that benefits from the sparse DVS output and the high dynamic range of the pixels. We have recorded activity such as mouse sleep cycles over periods of a week at millisecond resolution, in a data file of about 1GB size (<2kBps), although we have not yet published any results of these measurements. Here the low latency of the DVS has not been used, although it could allow feedback control.

Other applications we have recently explored include gesture recognition [22], whisker tracking, satellite tracking, yeast cell tracking in microfluidics, hydrodynamics with particle velocimetry [21], aerodynamics using soap bubbles, line following robots, and obstacle detection using a pulsed laser line.

Computing with Suspicious Coincidences

Although object tracking is natural and easy with the DVS, it is somehow limited by the lack of a full cortically-inspired hierarchy of computation. However even object tracking already takes advantage of spatio-temporal event occurrence: Moving objects emit events like the familiar sparklers waved around on holiday occasions. It is the spatio-temporal coincidences of these events that drive the tracker models. Vision is

[2] "PigTracker" comes from the line drawing of a pig used during development in Telluride.

often considered to be the process of object recognition. Now we observe from biology that there exists an impressive amount of cortical tissue that expands the visual representation of the dynamic visual input to a high dimensional representation. How can we bring these ideas into algorithmic processing of the retina output, while somehow taking advantage of the event-based output which affords us information about spatio-temporal coincidences in the sensor input?

I tried to instantiate some ideas about early feature extraction into two jAER classes, SimpleOrientationFilter and DirectionSelectiveFilter. The SimpleOrientationFilter expands the representation of events from the On/Off of the DVS output to add Orientation as another field of the output events. OrientationEvents are computed by measuring "suspicious coincidences" at a particular orientation. A spatial map of most-recent event times is used as input to the algorithm. An orientation event is only output if the events lying along a particular orientation are temporally coincident, as they would be if they were produced by a moving edge. This edge produces a plane with a cliff to past times in the spatial map of event times. This filter works robustly on scenes with clear edges like hands or indoor spaces, although we have not tried to quantify the performance. In any case, the next obvious step was to include another ubiquitous feature of cortical simple cells, that they are almost always direction selective. Therefore, DirectionSelectiveFilter takes packets of OrientationEvent as inputs, and outputs packets of MotionOrientationEvent. These events add "direction" and "speed" fields to the OrienationEvents and are computed using time-of-flight of OrientationEvents. These events are very noisy (as is generally the case with local motion computations) but by integrating them over translational, tangential, and radial directions we obtain a fairly robust measure of global optical flow. One possible next step was obviously binocular vision: By correctly correlating vertical orientation events from the two eyes we should be able to obtain some stereo binocular disparity information; in practice this works in artificial simple scenes but not yet in realistic natural scenarios. Ryad Benosman's group has made the most progress in full stereo vision [15,16], but personally I have only had convincing success in using stereo vision to binocularly track small moving objects like balls; see the jAER class StereoClusterTracker for details.

Of course the real aim here is to obtain the motion parallax flow that signifies scene structure from a moving camera. To this end, inspired by our friends in Sevilla [25], I recently integrated a 3-DOF rate gyro on the back of a DVS camera. This sensor provides independent measure of the camera rotation. By combining this camera rotation information with measured local optical flow, I hope we can robustly detect obstacles in the environment on a power budget more competitive with that of flying insects. This target has long been an aim of neuromorphic engineering and although we are not there yet, we are getting closer. Together with developments of new sensors, new hardware for sensor processing, and inventive new algorithms, we are sure to have a grand time over the next few years.

Acknowledgements. Our Sensors group has particularly benefitted from the work of the following individuals: Rodney Douglas, Shih-Chii Liu, Wolfgang Henggeler, Patrick Lichtsteiner, Raphael Berner, Christian Brandli, Kynan Eng, Holger Finger,

Peter O'Conner, and Minhao Yang. We gratefully acknowledge support via the EU projects CAVIAR and SEEBETTER, the Swiss National Science Foundation NCCR Robotics Project, the Samsung Advanced Inst. of Technology, the University of Zurich, and ETH Zurich. Many ideas and projects were first conceived at the Telluride Neuromorphic Cognition Engineering Workshop (http://neuromorphs.net) sponsored by the US National Science Foundation and at the Capo Caccia Cognitive Neuromorphic Engineering Workshop (http://capocaccia.ethz.ch), organized by Giacomo Indiveri and Rodney Douglas.

References

1. Mahowald, M.A.: An Analog VLSI System for Stereoscopic Vision. Kluwer, Boston (1994)
2. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 dB 15 μs latency asynchronous temporal contrast vision sensor. IEEE Journal of Solid-State Circuits 43(2), 566–576 (2008)
3. Dynamic Vision Sensor (DVS) - asynchronous temporal contrast silicon retina (2012), `http://siliconretina.ini.uzh.ch`
4. Conradt, J., Berner, R., Cook, M., Delbruck, T.: An Embedded AER Dynamic Vision Sensor for Low-Latency Pole Balancing. In: 5th IEEE Workshop on Embedded Computer Vision (in conjunction with ICCV 2009), Kyoto, Japan, pp. 1–6. IEEE (2009)
5. Delbruck, T.: Frame-free dynamic digital vision. In: Proceedings of Intl. Symp. on Secure-Life Electronics, Advanced Electronics for Quality Life and Society, Tokyo, University of Tokyo, pp. 21–26 (2008)
6. Delbruck, T., Lichtsteiner, P.: Fast sensory motor control based on event-based hybrid neuromorphic-procedural system. In: ISCAS 2007, New Orleans, pp. 845–848 (2007)
7. Berner, R., Delbruck, T.: Event-Based Pixel Sensitive to Changes of Color and Brightness. IEEE Transactions on Circuits and Systems I: Regular Papers 58(7), 1581–1590 (2011)
8. Delbruck, T., Linares-Barranco, B., Culurciello, E., Posch, C.: Activity-Driven, Event-Based Vision Sensors. In: IEEE International Symposium on Circuits and Systems, Paris, pp. 2426–2429 (2010)
9. Liu, S.C., Delbruck, T.: Neuromorphic sensory systems. Current Opinion in Neurobiology 20(3), 288–295 (2010)
10. Liu, S.C., van Schaik, A., Minch, B.A., Delbruck, T.: Event-based 64-channel binaural silicon cochlea with Q enhancement mechanisms. In: IEEE ISCAS 2009, pp. 2426–2429 (2010)
11. jAER open source project: Real time sensory-motor processing for event-based sensors and systems, (2007), `http://jaer.wiki.sourceforge.net`
12. Seebetter project (2011), `http://www.seebetter.eu`
13. Serrano-Gotarredona, R., Oster, M., et al.: CAVIAR: A 45k Neuron, 5M Synapse, 12G Connects/s AER Hardware Sensory–Processing– Learning–Actuating System for High-Speed Visual Object Recognition and Tracking. IEEE Trans. on Neural Networks 20(9), 1417–1438 (2009)
14. Abshire, P., et al.: Confession session: Learning from others mistakes. In: ISCAS 2011, Rio de Janeiro, pp. 1149–1162 (2011)

15. Rogister, P., Benosman, R., Ieng, S.-H., Lichtsteiner, P., Delbruck, T.: Asynchronous Event-Based Binocular Stereo Matching. IEEE Transactions on Neural Networks and Learning Systems 23(2), 347–353 (2012)
16. Benosman, R., Sio, H., Ieng, X., Rogister, P., Posch, C.: Asynchronous Event-Based Hebbian Epipolar Geometry. IEEE Transactions on Neural Networks 22(11), 1723–1734 (2011)
17. Matolin, D., Posch, C., Wohlgenannt, R.: True correlated double sampling and comparator design for time-based image sensors. In: ISCAS 2009 (2009)
18. Posch, C., Matolin, D., Wohlgenannt, R.: A QVGA 143dB dynamic range asynchronous address-event PWM dynamic image sensor with lossless pixel-level video compression. In: 2010 IEEE Solid-State Circuits Conference Digest of Technical Papers, ISSCC (2010)
19. Camunas-Mesa, L., Zamarreno-Ramos, C., Linares-Barranco, A., Acosta-Jimenez, A.J., Serrano-Gotarredona, T., Linares-Barranco, B.: An Event-Driven Multi-Kernel Convolution Processor Module for Event-Driven Vision Sensors. IEEE Journal of Solid-State Circuits 47(2), 504–517 (2012)
20. Landolt, O., Mitros, A., Koch, C.: Visual sensor with resolution enhancement by mechanical vibrations. In: Advanced Research in VLSI, 2001. Proceedings of ARVLSI 2001, pp. 249–264 (2001)
21. Drazen, D., Lichtsteiner, P., Hafliger, P., Delbruck, T., Jensen, A.: Toward real-time particle tracking using an event-based dynamic vision sensor. Experiments in Fluids 51(5), 1465–1469 (2011)
22. Lee, J., Delbruck, T., Park, P.K.J., Pfeiffer, M., Shin, C.W., Ryu, H., Kang, B.C.: Live demonstration: Gesture-Based remote control using stereo pair of dynamic vision sensors. In: ISCAS 2012, Seoul (in press, 2012)
23. Yang, M., Liu, S.C., Li, C., Delbruck, T.: Addressable Current Reference Array with 170dB Dynamic Range. In: ISCAS 2012, Seoul (in press, 2012)
24. Boahen, K.A.: A burst-mode word-serial address-event link-I transmitter design. IEEE Transactions on Circuits and Systems I-Regular Papers 51(7), 1269–1280 (2004)
25. Jimenez-Fernandez, A., Fuentes-del-Bosh, J.L., Paz-Vicente, R., Linares-Barranco, A., Jiménez, G.: Live demonstration: Neuro-inspired system for realtime vision tilt correction. In: ISCAS 2010, pp. 1393–1397 (2010)
26. Lenero-Bardallo, J.A., Serrano-Gotarredona, T., Linares-Barranco, B.: A 3.6 us Latency Asynchronous Frame-Free Event-Driven Dynamic-Vision-Sensor. IEEE Journal of Solid-State Circuits 46(6), 1443–1455 (2011)