CMOS Continuous-Time Selective Change Driven Vision Sensor

Fernando Pardo Jose A Boluda Francisco Vegara
Departmento de Informática
Universitat de València
46100 Burjassot, Valencia, SPAIN
Email: http://tapec.uv.es/

***Abstract*—Selective Change Driven (SCD) Vision consists of delivering, on demand, the pixel that has undergone the largest change since the last time it was read out. This implies a continuous pixel flow ordered depending on their change, which is usually related to the importance of each pixel in the implemented algorithm. This strategy reduces by several orders of magnitude the information sent by the vision sensor without losing accuracy, since only relevant information is sent to the processing hardware. The SCD sensor presented here is based on a continuous-time sensing cell and delivers its illumination level and its address with a minimum delay of 2 μs. At this speed, the sensor is able to signal events, along with its grey value, with a time resolution of 2 μs, allowing the real-time tracking of very fast moving objects.**

I. Introduction

Biological vision systems are the result of million years of evolution. In many aspects, they are not better than our artificial image systems, but they are the proof that a different approach can be useful to solve most vision problems. Pixels of the biological system work independently and each one sends its information asynchronously; it seems that there are no images or snapshots involved in the processing, just pulses and events coming from anywhere, of the focal plane, at any time. The processing data in biological systems seem to be the pixel flow instead of the image flow of conventional artificial vision systems.

Many artificial vision sensors have been inspired by biology and they send a flow of pixels instead of a flow of images. This idea was proposed for vision sensor design in the early nineties [1]. This strategy was called AER (Address Event Representation) because the only information that was sent by the sensor was the address of the pixel generating the visual event. This concept has evolved over the years and there are many sensors based on this strategy.

Some of the AER sensors are based on integration cells (a photo-generated current charge a capacitor, so the voltage of the capacitor depends on the light intensity and the elapsed time). Conventional vision cameras utilize a fixed time for integration, but most AER sensors employ a fixed threshold voltage so the intensity depends on the integrating elapsed time, generating a pulse train, which frequency directly depends on light intensity (like in most biological systems). This strategy resembles biological systems, but it has a time resolution of milliseconds (like the biological systems) which is very slow if it is compared with other artificial strategies.

Other AER sensors utilize a continuous transduction cell that transforms the photo-generated current in a voltage that can be read-out at any time. The time resolution of these *continuous-time* systems is reduced by at least two orders of magnitude compared to the integration based ones, though the visual information quality is not so good due to the low currents involved.

AER sensors can be also characterized depending on the function utilized to trigger the events. We find sensors that fire pixels depending on the illumination level, spatial contrast (differences of the pixel with its neighbors) and temporal contrast (change of the illumination the pixel has undergone during some interval). From a dynamic point view (motion processing), the most interesting function is the one based on temporal contrast, since it selects those pixels that have changed due to the scene movements. It also has the advantage of combining in one parameter, not only the temporal contrast, but also the spatial contrast (object edges) since those parts with higher spatial contrast will produce also higher temporal contrast.

There are many AER sensor realizations to date, so we just mention a few that present some special characteristics. For example, sensors [2] and [3] (both very similar) are extremely fast and are able to track 5-10 kHz moving objects. The sensor in [4] is able to work using three different functions to select events (illumination level, spatial and temporal contrast) with very few transistors; though the event time resolution is low, since it is based on fixed-time integration. The sensor in [5] has good event time resolution (continuous-time); it is one of the few AER sensors that provides good illumination level information, though at lower time resolution than the events, since illumination is based on integration (fixed voltage). The sensor in [6] is the only one that uses color change for firing events. The sensor described in [7] is one of the sensors that best models the human eye behavior. The sensor in [8] is foveated and it models the behavior of the inner foveal and outer retinal cells of the eye.

A. SCD Vision

Most of dynamic (based on temporal contrast) AER sensors do not provide illumination level information, but just the address of the pixel that produced the event. It is not possible to extract characteristics from the illumination information, and this is very important to understand the environment to process. There are some sensors that are able to send the illumination level information, but not at the fast time resolution of the events.

AER sensors do not usually order the events by relevance, and if they do, they are based on integration and have a slow event response and they do it asynchronously, which can be difficult for the hardware that has to process the information.

The SCD concept was first defined and introduced in [9] to overcome these problems. The basic working principle of any SCD vision sensor is simple:
Single pixel illumination level and address is sent out, upon external request, ordered by the difference in the illumination level that each pixel has undergone since the last time it was read-out.

This principle implies that pixels are sent ordered by their relevance from the dynamic point of view. The pixel flow is continuous, synchronized and mastered by the processing unit, which is in charge of processing the visual information and the best suited to fix the pixel rate required by the vision algorithm.

Any pixel can be sent in an SCD sensor if it is the most relevant at any time, even if the change has been small. This is a big difference with current AER sensors that send pixels only if they are above some threshold. Furthermore, in the AER selection function, the change is computed during a fixed small time, so pixels that have changed a lot, but during a long time, are not perceived by these sensors. SCD adapts itself to these situations by always providing a continuous flow.

The advantages of this strategy were first demonstrated by simulations [10], [11]. Afterwards, a 32×32 SCD sensor was designed using a cell based on fixed time integration [12], [13]. Event ordering was based on a Winner-Take-All (WTA) circuit of 1024 inputs with single winner selection [14]. This first version had a slightly large pixel access time of 10/20 μs and very low Signal to Noise Ratio (SNR). The effective integration time could not be lower than 500 μs limiting the event time resolution, so all benefits of SCD vision could only be partially met. Nevertheless, this first attempt already served to validate some of the benefits of the SCD strategy in real-time tracking experiments, outperforming by one to two orders of magnitude a conventional frame-based system [15], [16].

In this paper, we present a new version of a 64×64 SCD sensor based on a continuous-time logarithmic response cell and renewed Winner-Take-All (WTA) circuit and selection logic. The event ordering circuit is based on a 4096 input WTA able to select a single winner pixel in just 1 μs. The pixel address and its illumination level are presented at the output with a simple and synchronous slave mode interface, at an optimum minimum pixel latency of 2 μs for SNR reasons. The continuous-time reading photoreceptor allows for the detection and signaling of events in less than 2 μs, which is the same time resolution of a frame-based imager working at 500,000 fps.

II. BASIC PIXEL

The SCD sensor has an array of 64×64 pixels or sensing elements. Each pixel must detect and signal that it has undergone the largest change in illumination, since the last time it was read-out. Any pixel can detect if it is the winner (it has the largest change) thanks to the Winner-Take-All circuit

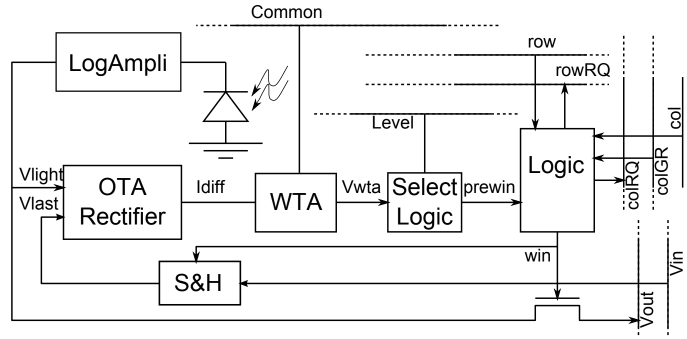

Fig. 1. Block diagram of a SCD single pixel

(WTA). All pixels participate in this WTA circuit sharing a single line (marked as $Common$ in Fig. 1). The WTA circuit usually chooses a single winner if few pixels have undergone large changes. Nevertheless, when there is a large amount of pixels with similar change magnitude, the WTA is unable to select just a single winner. In these cases, a distributed digital arbitration takes action and selects just a single pixel among the winner set.

Fig. 1 shows the block diagram of each cell in the sensor array.

The basic cell works as follows (see Fig. 1): photodiode transforms incident light into a small current. This current generates a voltage with a logarithmic dependency through a well-known log-amplifier configuration, which is based on a weak inversion transistor, negative feedback amplifier [17] and a source follower. The signal $Vlight$ is the logarithmic voltage of the incident light at any time (no integration).

The Sample and Hold Circuit (S&H) stores the last read-out value. It is based on a capacitor of 600 fF and a switch. The voltage at the capacitor is marked in the figure as $Vlast$. The magnitude compared among all pixels is the difference between the last read-out value $Vlast$, and the current incident light $Vlight$. This difference is calculated by means of an Operational Transconductance Amplifier and Rectifier (OTA Rectifier) that transforms the absolute difference of the current and last voltages into a current $Idiff$:

$$Idiff = G|Vlight - Vlast| \qquad (1)$$

where G is the transconductance factor expressed as A/V or Ω^{-1}.

The $Idiff$ currents of all pixels are compared thanks to the WTA circuit. The $Common$ line, shared by all WTA circuits in the sensor array, allows any pixel to generate a $Vwta$ that will be lower when larger is its $Idiff$ compared to all other $Idiff$ of the other pixels. The WTA is designed to pull-down the $Vwta$ of the winner pixel and pull-up the $Vwta$ of the rest, but as there are 4096 competitors, there can be more than one with a low voltage on this line indicating a winner condition.

The $Vwta$ voltage is then compared to a threshold level; if $Vwta$ is below this threshold, the pixel may think it is a winner and sets its $prewin$ signal. When there are few competitors, only one winner is clearly identified among them, and it has a $Vwta$ voltage that is usually always the same. To fix a threshold value in this case is very easy. When there are so many competitors in the WTA, several pixels could share the prize. The $Vwta$ of these winning pixels is different depending on the number of winners; the lowest $Vwta$ voltage is obtained when there is a single clear winner, and increases with the number of winners. The threshold $Level$, shared among all pixels, is dynamic in this circuit and adapts itself depending on the lowest $Vwta$ found in all pixels.

All pixels that have set their $prewin$ signal will enter in a competition to select just a single winner. This is done by the logic circuit and the digital shared column and row lines shown in Fig. 1. If $prewin$ is set, the logic circuits sets the column request $colRQ$ signal. In there are many winners there are probably many columns with the $colRQ$ activated. There is an arbitration logic (see Fig. 2) that grants just one of the columns, setting the $colGR$ of the selected column. The pixel detects this $colGR$ and then sets its row request $rowRQ$, because there could be more than one winner pixel cell in the selected column. All these row requests share an arbitration circuit that immediately decides a single row winner, thus a single array winner.

The single winner procedure explained before is called competition and takes place just in few nanoseconds. At the end of the competition, the row and column arbitration logic knows the winner, but they will wait until receiving the clock synchronization signal to inform the pixel. The external clock latches the column and row winners (col and row signals) so the winner will set its win signal when col and row are both set. The win internal signal shorts the switch in the Sample and Hold circuit, charging the capacitor to $Vlight$. Since now $Vlight = Vlast$ then $Idiff = 0$, and this pixel loses current competition and unsets $prewin$, though the col, row and win signals remain set until a new clock signal is received. The pixel drives the columns shared $Vout$ with its illumination level. It goes to a buffer at the end of the column and it is sent back through the shared Vin, so there is enough current from this buffer to charge the capacitor in a short time.

The logAmpli circuit has 5 transistors and a 6×6 μm^2 Nwell photodiode. The OTA rectifier is the largest circuit in the pixel with 15 transistors. The S&H has two transistors for the switch and a MIM capacitor that occupies more than half of the pixel, though it is above the other circuits. The WTA circuit takes just 3 transistors. The Logic Select circuit is a threshold programmable inverter with 5 transistors. Finally, the Logic circuit has one inverter, one NAND and three wired-or shared gates, which makes nine transistors. There is one switch at the shared output with two more transistors. The whole pixel has then been designed with 41 transistors and occupies 30×30 μm^2. Fill factor is 4%.

III. Global sensor components

Fig. 2 shows the global components of the sensor along with the pixel array already presented in the previous section. The Row and Col single winner select circuits are responsible for the single winner selection and the sensor access, as explained in the previous section. Both Row and Col circuits are the same, though the $rowGR$ output is left open in the Row single winner select circuit because it is not necessary. The Column out amplifiers circuit contains one unity gain buffer for each column. The selected column buffer charges the S&H capacitor of the selected pixel and serves as the primary stage for the single output amplifier. Column/Row encoders/decoders, translate the 6-bit addresses of the chip to the 64 selection lines and vice versa.

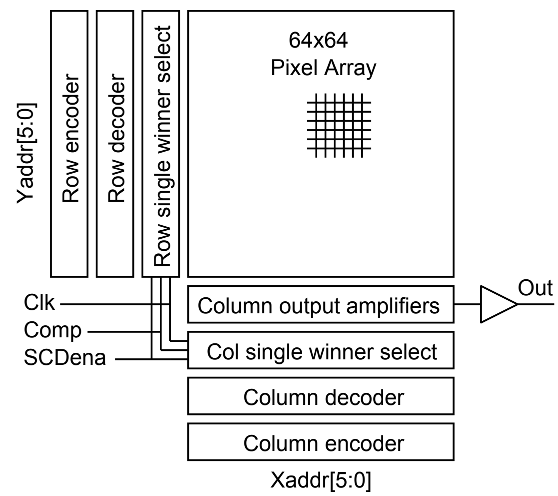

Fig. 2. Block diagram of the complete SCD sensor

This SCD sensor can work in two modes. The SCD mode is the mode the chip has been designed for. In this mode, the sensor drives the lines to send the address of the selected pixel (the most interesting). In non-SCD mode, or conventional mode, the sensor can be set to work as a standard imager. In this conventional mode, the address is driven by the external processor, which sets the position of the pixel to be read out. In this mode, the sensor acts like a RAM from which the processor reads the illumination values of each pixel it addresses. The mode is set by the $SCDena$ (SCD enable) signal and the mode change takes just few nanoseconds, so it can be set for every single pixel if desired. The address bus (formed by $Xaddr[5:0]$ and $Yaddr[5:0]$) is bidirectional, so the $SCDena$ signal controls both the mode and the direction of the address bus.

Fig. 2 just shows the signals that control sensor operation. There are some other signals necessary for biasing all the internal circuits in the sensor. There are nine bias signal to set all internal currents and voltages. All these bias inputs require a constant voltage that is translated to current if necessary.

Fig. 3 shows a photograph of the SCD sensor die. The sensor has been fabricated with the AMS 180 nm CMOS process. The technology is single poly but it has the possibility of MIM (Metal-Insulator-Metal) which allows for the inclusion of large capacitors on top of other circuitry. The 64×64 pixel array takes an area of 1.92×1.92 mm^2 while the whole sensor is a square of 2.53×2.53 mm^2 including pads, so the pixel array takes most of the core space.

IV. Sensor Camera and operation

The SCD sensor has been designed to have a very simple interface and operation with an external microprocessor or microcontroller. Table I summarizes all inputs/outputs of the SCD chip.

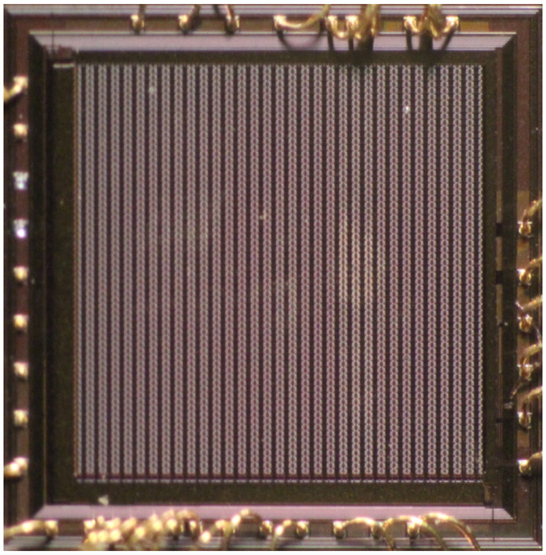

Fig. 3. Photograph of the fabricated SCD sensor

TABLE I. INPUTS AND OUTPUTS OF THE SCD SENSOR CHIP

Signal	Function
Vdd, AVdd	Digital and Analog supplies (1.8 V)
GND, AGND	Digital and Analog grounds
Out	Illumination level analog out
Xaddr[5:0]	Column address lines
Yaddr[5:0]	Row address lines
Clk	Synchronization clock
Comp	Competition signal
SCDena	SCD mode (1=SCD, 0=Conventional)
Bias[8:0]	Bias signals

The sensor has separate power supplies for analog and digital to avoid digital noise in the analog part. Power supply is 1.8 volts. All digital lines are also 1.8 V logic. There are nine bias signals, which have to be programmed to a specific voltage. The interface with the microcontroller is a simple RAM-like interface with twelve address lines ($Xaddr[5:0]$, $Yaddr[5:0]$) one analog Out, which requires an A/D converter (ADC) and the clock signal Clk for synchronization. $Comp$ is the competition signal and can be derived from Clk. $Comp$ can be exactly as Clk, but it is better to make it wider than the Clk. In conventional mode, $Comp$ can be tight down permanently.

Fig. 4 shows a diagram of the designed camera for testing and measure the sensor. The camera is based on a Microchip PIC32MZ microcontroller. This microcontroller runs at 200 MHz and incorporates a 1 μs 10-bit ADC and a USB interface. The analog bias voltages are generated using several 8-bit DAC (Digital to Analog Converters) encapsulated in a couple of chips programmed using a serial I2C interface. The microcontroller has 3.3 V logic, thus there are logic level translators between the SCD sensor and the microcontroller (not shown in the figure). These level translators are located in the middle of the address and control lines.

Fig. 5 shows the synchronization diagram during typical SCD and non SCD operation. During SCD operation, the processors drives the $SCDena$ signal high. When the micro is ready to process a new pixel, it asserts the signal $Comp$ and then Clk. At this moment, a competition starts in the SCD sensor. The competition takes no more than 50 ns, so

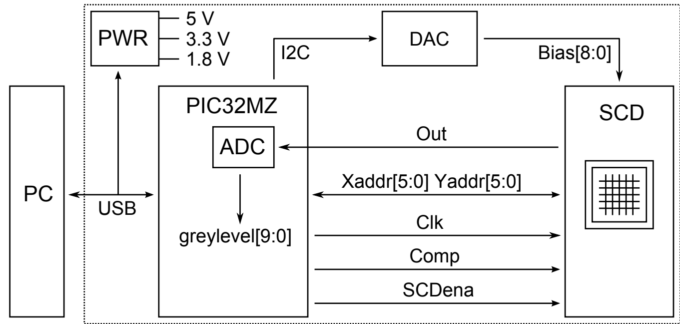

Fig. 4. Camera block diagram

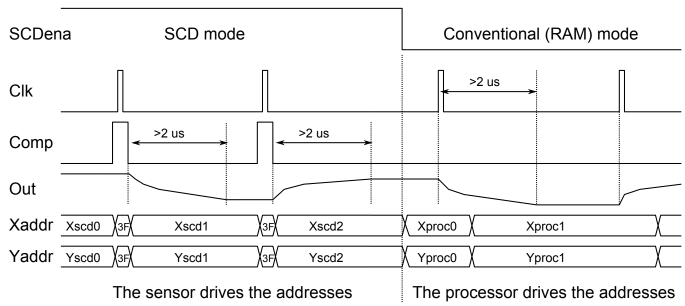

Fig. 5. Time diagram of SCD and conventional modes of operation

the processor can set $Comp$ low after that time. The only condition for the Clk signal is that it must be narrower that the $Comp$ pulse. During the $Comp$ pulse, the sensor drives the address lines high because they are disconnected from the inner Col and Row signals, so no pixel is selected at this time. When the Clk goes low, the new winning address is stored and when $Comp$ goes low this stored address is propagated to the inner Col and Row lines selecting a new pixel. At the same time, Col and Row also generate the address that is seen immediately in the external address bus. At this moment, the illumination level of the selected pixel is propagated to the output and to the selected pixel, charging the internal capacitor to this value. The settle time of this analog output takes less than 2 μs. After this time, the microcontroller ADC can start conversion. In addition, the microcontroller can start a new pixel read-out by repeating the same sequence.

Fig. 5 also shows the timing for the conventional operation. In non-SCD mode, or conventional operation, the sensor behaves as a synchronous access RAM; the processor puts the address and the sensor returns the data corresponding to the addressed pixel (in this case, the analog illumination value of that pixel) using a clock to latch the address. During conventional operation, the sensor first puts the address and then generates a pulse in the clock so the sensor can latch that address. Right from the rising edge of the clock, the pixel is selected and its corresponding output takes less than 2 μs to settle, as seen on the timing diagram. The microcontroller can change the address lines at any time between clock pulses; minimum clock pulse width, and hold and set-up times can be as small as few nanoseconds, so the only parameter limiting the maximum frequency operation is the 2 μs analog output settling time.

In both modes, the microcontroller always drives the con-

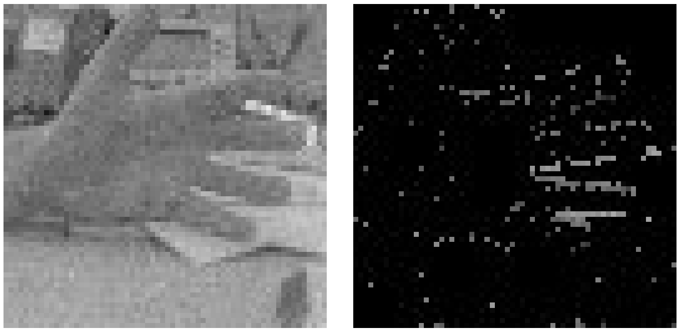

Fig. 6. Image taken with the SCD camera: Conventional mode (left), SCD mode (right)

trol signals and fixes the pixel rate, so it can adapt this rate to the special needs of the implemented image-processing task. The maximum pixel rate for a reasonable quality is 500,000 pixels per second (2 μs period).

V. Sensor test and results

The sensor has been tested using the camera described in the previous section. In conventional mode, the microcontroller firmware send images to the PC software at a programmable rate. In SCD mode, the firmware sends the last programmable number of pixels captured by the sensor. The PC software has a window to show a reconstruction of the image by holding the SCD values received, or to show just the received pixels from the sensor. In this last case, only those pixels with the largest change are sent, so there is only a portion of the image at that instant (the most interesting).

Fig. 6 shows a complete image (left) and a partial image as obtained using the last read pixels in SCD mode, while moving a hand in front of the camera. The complete image is obtained in conventional mode or in SCD mode but holding the past results and moving the camera so all pixels register a change. It can be understood that the SCD sensor automatically compresses the image by sending just the changing pixels.

The continuous-time sensing cell can run very fast since it is not based on light integration. Nevertheless, the noise at the output is very high as shown in Fig. 6. This low SNR reduces the effective grey level resolution to 6-bits, which is very low for some applications, but good enough for many others, especially for high-speed movement analysis.

This continuous-time sensing cell also presents a very high Fixed Pattern Noise (FPN). This FPN can be easily cancelled capturing a background image and then subtracting it in normal operation. It is done inside the microcontroller during the capture process so the image comes clean to the PC. This high FPN is not a problem inside the SCD sensor, because the focal processing only requires a comparison of each pixel with itself, so no matter there is a high FPN because it is automatically cancelled in the comparison.

The sensor has not been designed for low power consumption and it employs relatively large currents to bias most of the inner circuits. Nonetheless, power requirement is below 60 mW, which is one order of magnitude less than the full power required by most 32-bit microcontrollers. In fact, the entire camera, including microcontroller, SCD sensor, DACs, level shifters, etc., is supplied with just the USB cable connected to the PC. No extra power supply is required.

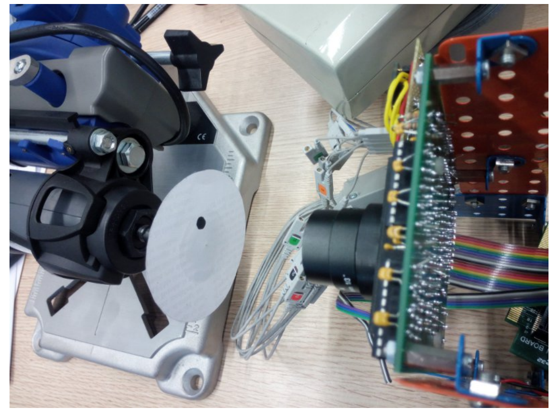

Fig. 7. Top view of the tracking experiment set-up: fast rotating tool (left), SCD camera (right)

A. Tracking experiment

One of the best applications of event-based vision sensors is the tracking of fast moving objects. The on-chip selection of moving pixels reduces the amount of data to process and enhances the time resolution of the movement sampling. The SCD sensor opens the possibility of tracking fast objects, in real-time, with just a simple microcontroller. Other approaches to solve this same problem would require a conventional camera running at more than 10,000 fps, with a multipath/multicore system to record and compute the track.

Fig. 7 shows a photograph of the top-view experiment set-up. It consists of a rotary tool (shown at left of the photograph) with a white circle containing a black spot. Based on the tool specifications, it runs from 10,000 to 33,000 rpm with no load. At the right of the photograph, we can see the SCD camera with a 4 mm, 1/3″ lens. The camera is powered just with the USB cable.

The microcontroller in the camera has been programmed to capture 1024 pixels at a constant rate of 3.5 μs/px. The total recorded time is thus 3.5 ms approximately, which is already less than the 20 ms inter-frame time of any video sequence working at 50 fps. A conventional camera would have only taken one picture during this time and no movement is possible to extract from just one picture.

Fig. 8 shows the X and Y coordinates of the received pixel belonging to the rotating spot at any instant. The grey level is employed to detect if a pixel comes from the moving object. In this first experiment, the rotary tool is working at its minimum speed (supposedly 10,000 rpm). The figure shows two clear sinusoidal functions, dephased by 90 degrees, describing a circumference track. The dispersion shown in the plots is due to the size of the moving spot, which is roughly eight pixels wide. The period of the signal is approximately 4.8 ms, which corresponds to a rotating speed of 12,500 rpm.

Fig. 9 shows the results when the rotary tool is at its maximum speed. Again, the circumference path is accurately tracked. In this case the signal period is roughly 2.25 ms and the resulting rotating speed is about 26,700 rpm. This speed is

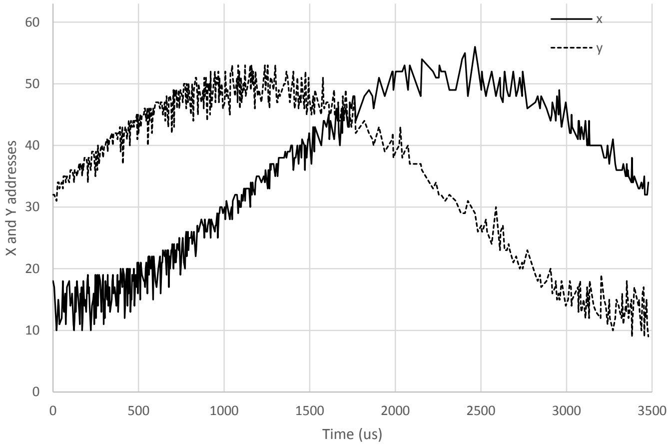

Fig. 8. X and Y coordinates of 1024 pixels read in SCD mode during 3.5 ms (tool at minimum speed)

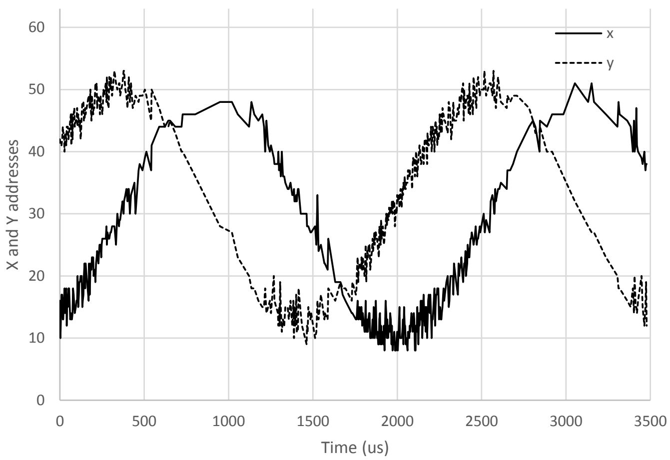

Fig. 9. X and Y coordinates of 1024 pixels read in SCD mode during 3.5 ms (tool at maximum speed)

lower than the tool specifications, probably due to the attached load.

These experiments have shown that it is possible to track a fast moving object with a time resolution close to the pixel acquisition time (in the order of microseconds). The same results would require a conventional high-speed camera working at 100,000 fps or more, and the processing of such a huge amount of data would require a processing hardware and data path accordingly scaled.

VI. Conclusion

A new Selective Change Driven Vision sensor has been presented. It is similar to AER sensors, but it provides illumination information with the same time resolution than the events. Other advantages include the event ordering according to the illumination change; the completely synchronous and simple interface, mastered by the processing unit; pixel rate adaptability; conventional RAM mode of operation, etc. A simple fast tracking experiment has been included to show that this sensor can solve tracking problems that cannot be addressed with conventional hardware, even using the highest performance products, which anyway would increase system cost and would require high power consumption and space.

Acknowledgment

This work has been supported by Spanish Government project TEC2009-12980.

References

[1] M. A. Mahowald, "VLSI analogs of neural visual processing: a synthesis of form and function," Ph.D. dissertation, California Institute of Technology, Pasadena, California, May 1992.

[2] T. Serrano-Gotarredona and B. Linares-Barranco, "A 128x128 1.5% contrast sensitivity 0.9% FPN 3 μs latency 4 mW asynchronous frame-free dynamic vision sensor using transimpedance preamplifiers," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 3, pp. 827–838, Mar. 2013.

[3] J. A. Leñero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 3.6 μs latency asynchronous frame-free event-driven dynamic-vision-sensor," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 6, pp. 1443–1455, Jun. 2011.

[4] D. Kim and E. Culurciello, "Tri-mode smart vision sensor with 11-transistors/pixel for wireless sensor networks," *IEEE Sensors Journal*, vol. 13, no. 6, pp. 2102–2108, Jun. 2013.

[5] C. Posch, D. Matolin, and R. Wohlgenannt, "A QVGA 143 dB dynamic range frame-free PWM image sensor with lossless pixel-level video compression and time-domain CDS," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 259–275, Jan. 2011.

[6] R. Berner and T. Delbruck, "Event-based pixel sensitive to changes of color and brightness," *IEEE Transactions on Circuits and Systems*, vol. 58, no. 7, pp. 1581–1590, Jul. 2011.

[7] K. A. Zaghloul and K. Boahen, "Optic nerve signals in a neuromorphic chip II: Testing and results," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 4, pp. 667–675, Apr. 2004.

[8] M. Azadmehr, J. P. Abrahamsen, and P. Häiger, "A foveated AER imager chip," in *IEEE Int. Symposium on Circuits and Systems, ISCAS'05*, vol. 3, Kobe, Japan, May 2005, pp. 2751–2754.

[9] F. Pardo, X. Benavent, J. A. Boluda, and F. Vegara, "Selective change-driven image processing for high-speed motion estimation," in *13th International Conference on Systems, Signals and Image Processing (IWSSIP'06)*, Sep. 2006, pp. 163–166.

[10] F. Pardo, J. A. Boluda, F. Vegara, and P. Zuccarello, "On the advantages of asynchronous pixel reading and processing for high-speed motion estimation," *Springer Lecture Notes in Computer Science (ISVC 2008)*, vol. 5358, pp. 205–215, Dec. 2008.

[11] J. A. Boluda, F. Vegara, F. Pardo, and P. Zuccarello, "Selective change-driven image processing: a speeding-up strategy," *Springer Lecture Notes in Computer Science (CIARP 2009)*, vol. 5856, pp. 37–44, Nov. 2009.

[12] P. Zuccarello, F. Pardo, A. de la Plaza, and J. A. Boluda, "A 32x32 pixels vision sensor for selective change driven readout strategy," in *Fringe Posters of the 36th European Solid-State Circuits Conference (ESSCIRC 2010)*, Sep. 2010.

[13] P. Zuccarello, F. Pardo, and F. Vegara, "Silicon implementation of a 32x32-pixel vision sensor for selective change driven read-out strategy," in *8th Conference on Ph.D Research in Microelectronics and Electronics (PRIME 2012)*, Jun. 2012, pp. 1–4.

[14] P. Zuccarello, F. Pardo, A. de la Plaza, and J. A. Boluda, "32x32 winner-take-all matrix with single winner selection," *Electronics Letters*, vol. 46, no. 5, pp. 333–335, Mar. 2010.

[15] F. Pardo, P. Zuccarello, J. A. Boluda, and F. Vegara, "Advantages of selective change-driven vision for resource-limited systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1415–1423, Oct. 2011.

[16] F. Vegara, P. Zuccarello, J. A. Boluda, and F. Pardo, "Taking advantage of selective change driven processing for 3D scanning," *Sensors*, vol. 13, no. 10, pp. 13 143–13 162, Oct. 2013.

[17] T. Delbruck and C. A. Mead, "Adaptive photoreceptor with wide dynamic range," in *Int. Symposium on Circuits and Systems, ISCAS'94*, vol. 4, London, UK, 1994, pp. 339–342.