A Low-Power Smart Camera for Video Surveillance and Forensic Applications

M. Lecca[1], Y. Zou[1], Member, IEEE, S.S. Zurriaga[2], J.L.D. Orozco[2], and M. Gottardi[1], Member, IEEE

[1]Fondazione Bruno Kessler, 38122 Povo, Trento, Italy. e-mail: (lecca,zou,gottardi)@fbk.eu.

[2]Valencia Local Police, Av Del Cid 37, Valencia 46018, Spain. e-mail: proyectosplv@valencia.es.

***Abstract*— We describe a custom, low-power surveillance camera for the control of isolated, peripheral areas, that are often chosen for illegal activities, like traffic of drugs, weapons and cigarettes. The camera embeds on chip an image processing algorithm for real-time event detection, exploiting a dynamic background subtraction approach to identify the so-called *hot pixels*, i.e. pixels undergone an intensity variation, and possibly corresponding to an abnormal action. Every time a group of hot-pixels is detected, the camera generates an alarm that can be sent to an external processor for further processing (e.g. scene classification) or to a surveillance officer to plan an intervention. The related frames are saved to be used as forensic evidence in a court. The presented vision sensor combines the image array with a bench of processors, frame buffer, timing control block and digital interface. Low-power performance are obtained through custom chip design techniques, combined with integrated image processing and different sensor operating modes to minimize the sensor average power consumption. The vision sensor performance has been evaluated by two European LEAs, which expressed a high level of interest in such a system and a positive vote about its event detection performance.**

Keywords—Video-surveillance, event detection, dynamic background subtraction, smart sensor, low-power sensor.

I. Introduction

Surveillance systems are important tools to monitor indoor and outdoor environments, with the purpose of preventing crimes, and bringing criminals to the justice. Nevertheless, the surveillance systems currently adopted by the Law Enforcement Agencies (LEAs) are often bulky and power hungry. In fact, they usually employ commercial cameras, that continuously acquire videos of the observed area and send them to an external processor for further processing (e.g. extraction of features relevant to classify the acquired events) or to a surveillance officer, which performs a manual, often boring and error-prone, control. Moreover, these systems need to be supplied by an electricity infrastructure and require high power and bandwidth to record and transfer data. The amount of data delivered by these cameras is huge and generally redundant when compared with their final task, i.e. detection of anomalous actions. Therefore, the use of surveillance systems employing standard cameras is limited by (1) high power consumption; (2) the expensive infrastructure; (3) large amount of data to be transmitted. Smart cameras [1] might offer several advantages against a standard approach: low-power consumption thus long lasting operation; embedded image processing (e.g. edge extraction, histogram computation, binary classification of visual signal) [2]- [3]. These characteristics allow low-level image analysis to be performed at the early stage of the system (i.e. very close to the sensor). They would drastically reduce the amount of data to be transferred to the processor turning into an energy efficient system.

In this work, we describe the general hardware architecture of a low-power surveillance smart camera, implementing a real-time event detection algorithm [6]. The smart camera is always ON. It continuously acquires images of the observed scene in a grayscale, QVGA (320x240 pixels) format, down-samples them to QQVGA (160x120 pixels) format, and processes each QQVGA frame to detect *hot pixels*, i.e. pixels whose intensity value varies during time and that may correspond to anomalous activities. The intensity variations are detected by comparing the pixel intensity with two thresholds, that are specific for each pixel and that are dynamically updated based on the pixel intensity (thus they are self-regulating). Hot-pixels are first filtered through a topological erosion removing noise, then they are projected along the horizontal (x) and vertical (y) axes of the image. Projections are binarized against two user-defined thresholds, filtered to remove small, irrelevant *hot-pixel* aggregations, and finally used to generate an alert: if both the x- and y- projections are not null, the camera sends an alert to the processor.

The performance of the proposed smart camera has been assessed on a set of videos, simulating criminal actions. These videos have been processed by a Sensor Emulator, reproducing the sensor functionalities. The output has been evaluated by a group of LEAs surveillance officers through a questionnaire that aimed at measuring (1) the interest of LEAs in such a system as a tool for supporting the investigative and forensic activity, and (2) the LEA satisfaction about the event detection accuracy and about the usability of the information provided by the camera as forensic evidences. The results showed high interest and high satisfaction for the proposed technology.

The smart camera prototype is currently under manufacturing. A first sensor prototype with reduced functionalities is already available and fully tested. It integrates the processors for hot-pixel detection and delivers both, a QVGA 8 bit grayscale image and a quarter QVGA (QQVGA) hot-pixel bitmap.

II. Event-Based Approach

The main peculiarity of event-based vision application is that an event is not predictable and rarely happens. Under these assumptions, an event detection system, which in our case is based on a vision sensor connected to a processor, works as follows:

- **no detected event**: the sensor continuously analyses the scene without delivering any data, while the rest of the system is in idle-mode, consuming very low-power;
- **detected event**: the sensor provides information on the detected event, such as the position of the event inside the scene or a region of interest, and delivers images or short videos to the external processor for further processing.

This approach allows efficient energy management, which is implemented through a hierarchical system architecture, made of three layers (*Fig. 1*).

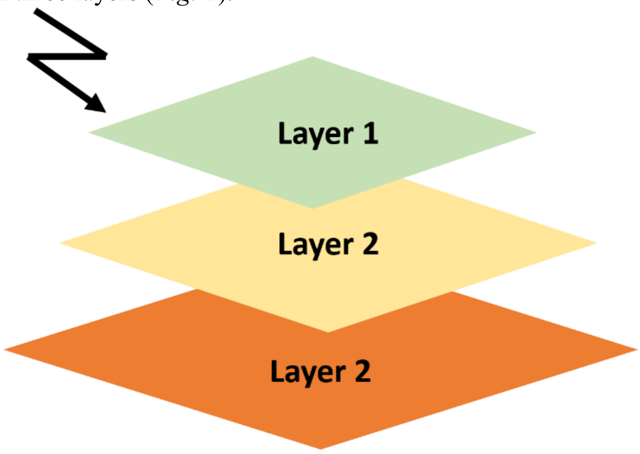

Fig. 1: Hierarchical approach for system energy-management. Layer 1: Sensing Layer with low-level image processing and event detection (generates a trigger for Layer 2); Layer 2: Processing Layer with high-level signal analysis, e.g. object/people/action recognition and classification; Layer 3: Communication Layer - data broadcasting. Layer 1 works continuously (100%) looking for events, while Layer 2 is usually OFF and wakes up upon Layer 1 request. Layer 3 is normally OFF and starts working upon Layer 2 request.

Layer 1 - Sensing: the vision sensor works continuously, acquires images and executes low-level visual processing in order to detect low-level events and to take simple decisions (e.g. generating a trigger for the processor);

Layer 2 - Processing: the processing layer is usually OFF and is woken up by Layer 1 as soon as it detects a low-level event. In this case, a high-level image processing is required and a specialized hardware is needed.

The power consumption of Layer 2 is much larger than one of Layer 1. This is the reason why Layer 2 is switched on only upon request. Its duty-cycle might range between 3% and 5% or even less in surveillance applications;

Layer 3 - Communication: data are sent to an external device. Wireless communication is one of the most power hungry activities of the system, requiring tens to hundreds of mW. Therefore, its use has to be minimized (reducing data bandwidth and duty-cycle). One strategy to overcome this problem is to send only symbolic information, instead of images. This information will be broadcasted only after the event has been properly analyzed and recognized as an alert.

This means that some image processing has to be accomplished locally instead of demanding all the computation to a remote server.

III. The Hardware-Oriented Event Detection Algorithm

The event detection is performed by the proprietary algorithm [4] that processes any input image in search of pixels whose intensity varies during time. These pixels, that are named *hot*, are computed by means of a self-regulating background subtraction technique and they possibly represent the events to be detected. The term “self-regulating” indicates that the intensity variation at each pixel is determined by the comparison of the pixel intensity signal with thresholds that are specific for that pixels and that are dynamically updated based on the pixel intensity at each frame.

Background subtraction is a very popular technique to detect event and it has been employed in several applicative scenarios, e.g. video-surveillance, life-assisted living, traffic monitoring [5] [6] [7] [8] [9] [10].

The algorithm [4] has been already employed in energy efficient systems [11], e.g. and it can be adapted to be low-power and CMOS compliant.

The final, hardware-oriented event detection algorithm workflow is sketched in Fig. 2 and consists of four main steps: (1) hot pixel computation; (2) noise removal; (3) x- and y-projections of the hot pixels and their binarization; (4) alert generation.

A. Hot Pixel Computation

The algorithm processes all the frames of any input video stream. Each frame is acquired as a gray level image with QVGA resolution, i.e. the size is 320 x 240 pixels. The frame is down-sampled to QQVGA format, i.e. the size of the down-sampled image is 160 x 120. The hot pixel detection is carried out on the down-sampled frame, in order to speed up the execution time and save energy.

The hot pixels are computed as follows. Let V_i be the ith frame after down-sampling. For each pixel x of V_i, the algorithm compares the intensity level $V_i(x)$ with two thresholds $V_i^m(x)$ and $V_i^M(x)$, that - as suggested by the notation - are specific for each frame and for each pixel x. Thus, the hot-pixel detection is based on a self-regulating thresholding strategy. If $V_i(x)$ falls out of the interval $\Gamma = [V_i^m(x) - \Delta_{HOT}, V_i^M(x), + \Delta_{HOT}]$, then x is labeled as *a hot pixel*, i.e. it may correspond to an event of interest. Otherwise, it is said to be *a cold pixel*.

The parameter Δ_{HOT} is an integer number and it is an user input. The thresholds $V_i^m(x)$ and $V_i^M(x)$ are positive real-valued numbers with $V_i^M(x) > V_i^m(x)$ for any pixel x. At the first frame V_0 the sensor does not perform any event detection, but uses the acquired data to initialize the values of the thresholds $V_i^m(x)$ and $V_i^M(x)$, precisely:

$$V_1^m(x) = V_0(x) - \Delta_{HOT}$$
$$V_1^M(x) = V_0(x) + \Delta_{HOT}.$$

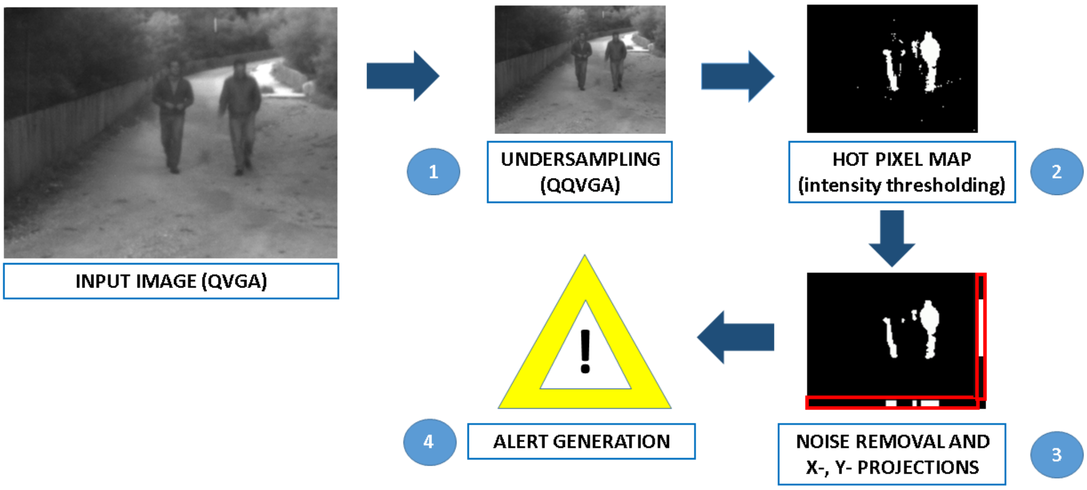

Fig. 2: Workflow of the event detection algorithm.

For each frame V_i with $i \geq 1$, and for each pixel x, the algorithm checks the membership of the intensity level $V_i(x)$ to the interval $[V_i^m(x) - \Delta_{HOT}, V_i^M(x), + \Delta_{HOT}]$ and updates the thresholds as follows:

- Threshold V_i^M:
  1) if $V_i^M(x) \leq V_i(x)$ then
  $$V_{i+1}^M(x) = V_i^M(x) + \Delta_{OPEN}; \tag{1}$$
  2) if $V_i^M(x) > V_i(x)$ then
  $$V_{i+1}^M(x) = V_i^M(x) - \Delta_{CLOSE}; \tag{2}$$
- Threshold V_i^m:
  1) if $V_i^m(x) \geq V_i(x)$ then
  $$V_{i+1}^m(x) = V_i^m(x) - \Delta_{OPEN}; \tag{3}$$
  2) if $V_i^m(x) < V_i(x)$ then
  $$V_{i+1}^m(x) = V_i^m(x) + \Delta_{CLOSE}. \tag{4}$$

The parameters Δ_{OPEN} and Δ_{CLOSE} are real-valued numbers, input by the user, with the constraint $\Delta_{OPEN} > \Delta_{CLOSE}$. Specifically, Δ_{OPEN} (Δ_{CLOSE}, resp.) is the increment (decrement, resp.) applied to $V_i^M(x)$ ($V_i^m(x)$, resp.) when the signal $V_i(x)$ is out of the range Γ. The parameter Δ_{CLOSE} is subtracted (added, resp.) to $V_i^M(x)$ ($V_i^m(x)$, resp.) when $V_i(x)$ is inside Γ.

Fig. 3 shows an example of a pixel changing its intensity over 16 frames. At 2nd frame, $V_{pix} := V_2(x)$ increases abruptly above $V_{max} := V_1^M(x)$, in particular $V_{pix} - V_{max} = d2 > 0$. Consequently, V_{max} is increased by Δ_{OPEN} at every new frame until V_{pix} returns back inside the gray region. Moreover, since between 2nd frame and 4th the distance $|V_{pix} - V_{max}|$ exceeds Δ_{HOT}, x is labeled as a hot-pixel. A complementary situation occurs during frames 13 and 14, where an hot-pixel occurs versus V_{13}^m. We observe that, in a steady state condition, the two thresholds V^m and V^M periodically switch between the conditions described by the Equations (1),(3) and (2),(4), without exceeding Δ_{HOT}.This period depends on the values of Δ_{OPEN} and Δ_{CLOSE}, that usually are chosen upon the applicative scenario. In the example discussed here, Δ_{OPEN} = 6LSB, while Δ_{CLOSE} = 1LSB, so that the periodicity is 7 frames. Setting up the optimal values of the algorithm parameters definitely depends on the task and on the general scene characteristics. In particular, under low-light conditions, the pixels usually do not undergo large variations, thus Δ_{OPEN} and Δ_{CLOSE} should be kept small. If the scene continuously varies, i.e. it is highly dynamic, the values of Δ_{OPEN} and Δ_{CLOSE} should be kept large, in order to reduce persistency effect (i.e. a pixel remains hot for long time). Finally, the smaller the value of Δ_{HOT}, the higher the sensitivity to intensity variation is.

Fig. 4 shows the algorithm behaviour under a periodic optical stimulus. Here, the role of the thresholds Δ_{OPEN} and Δ_{CLOSE} is even more evident and consists of inhibiting any signal variations according with their time response. Any signal variation occurred at a rate much lower or much larger than the rate at which the values $V_i^m(x)$ and $V_i^M(x)$ can change, after some frames is suppressed. In this case, the signal rate is much larger than the time response of V_{max} and V_{min}. Therefore, after a certain number of frames (about 100 in this example) the pixel reaches a new steady-state condition. At the higher level, this pixel does not contribute to the event detection phase anymore. Nevertheless, the fact that the pixel is no more hot does not necessary mean that the scene is static.

The output of this step is a binary QQVGA image, called the *hot pixel bitmap*: the hot pixels are displayed in white color, while the rest, i.e. the static background, is depicted in black color.

B. *Noise Removal*

Isolated hot pixels often do not correspond to event of interest, but they are mainly due to pixel saturation or to noise in scene acquisition. Since they are irrelevant, they are filtered by a topological procedure.

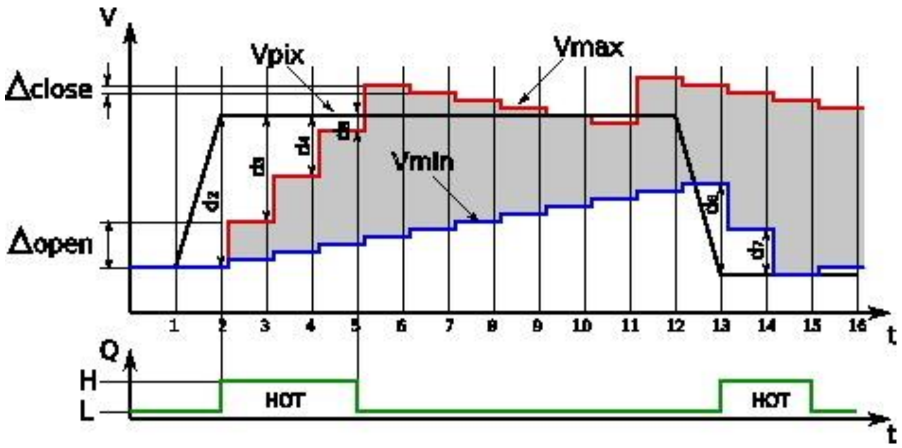

Fig. 3: An example of hot pixel detection. Vpix (in black) is the current photo-generated voltage. Vmax (in red) and Vmin (in blue) are the two threshold voltages used to determine the hot pixel condition.

Experiments showed that in general a standard morphological erosion filter may be too much aggressive, thus we implemented a slight erosion function, that works as follows. Let H denote a hot-pixel map; a hot-pixel x of H is retained if it is adjacent to at least b hot-pixels, where the threshold b is an integer, positive number, ranging over $\{0, \ldots, 8\}$. If b = 0, no erosion is performed and thus isolated hot-pixels are retained. If b = 8, our erosion function equals to the standard erosion morphological operator shrinking a region by one pixel.

C. X- and Y- Projections and Binarization

The hot pixels retained after the noise removal step are projected along the horizontal (x-) and vertical (y-) axes of the image. This operation is implemented for two reasons: (1) to perform a further filtering of the data; (2) to decide whenever to generate an alarm or not (i.e. weak up the external processor and start delivering data and/or advise a surveillance officer).

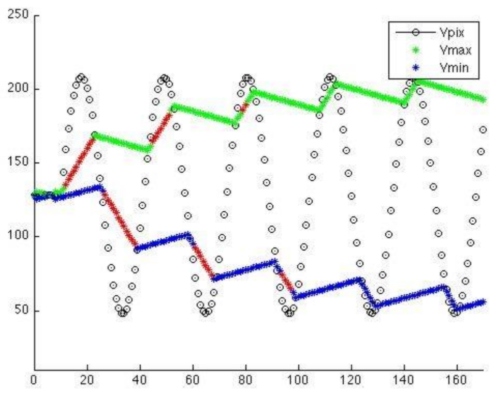

Fig. 4*:* Event detection algorithm in the case of a pixel stimulated periodically by a light source. The x-axis represents the number of frames, while y-axis is the signal amplitude coded into a 8-bit. After 10 frames of steady-state conditions, V_{pix} (i.e. the light impinging on the photodiode) starts oscillating with an amplitude of 80 DN (Digital Numbers) and with a period covering 30 frames. The red segments identify the frames where hot-pixels conditions occur, either on $V_{max} = V^M$ or $V_{min} = V^m$. When hot pixels are detected, V_{max} (V_{min}) is increased (decreased) by Δ_{OPEN}. In case of cold-pixels, V_{max} (V_{min}) is decreased (increased) by Δ_{CLOSE}.

Precisely, the horizontal and vertical projections of any hot-pixel map H are defined as two 1D vectors p_h and p_v with size N_c and N_r respectively, where N_c and N_r indicate the numbers of columns and rows of the image H. The entries of p_h (p_h, resp.) range over $\{0, \ldots, N_r\}$ ($\{0, \ldots, N_c\}$, resp.).

The projections are binarized as follows. Any entry of p_h (p_h, resp.) with value smaller than T_h (T_v, resp.) is cast to 0, otherwise it is cast to 1. The thresholds T_h and T_v (both integer numbers, strictly greater than zero) are user parameters that define the minimum and maximum linear size of the event to be detected. As the other thresholds of the algorithm, the optimal values of T_h and T_v must be fixed according to the applicative scenario (e.g. people or car detection) and to the geometric constraints of the camera (e.g. its position and focus).

If p_h and/or p_v after binarization are identically null, then no alarm is generated. Otherwise, an alert is delivered to an external processor and/or to a surveillance officer, along with the gray level QVGA image, the hot pixel map and the binarized projections.

IV. Sensor Architecture

Fig. 5 shows the block diagram of the sensor architecture. The array of pixels has a QVGA format, while the algorithm runs on a QQVGA image. The chip consists of 320 Column-level Amplifiers; an array of 160, 8bit column-level Processors, for parallel-wise row-selected pixel computation; a 10bit SRAM, used as frame buffer to store the threshold values $V^m(x)$ and $V^M(x)$ for each pixel; a Control Register storing the algorithm parameters ($\Delta_{OPEN}, \Delta_{CLOSE}, \Delta_{HOT},$), thresholds (T_h, T_v) and type of erosion filter; a Controller managing the sensor interface.

V. Experiments

In this Section, we report our experiments, aimed at evaluating: (1) the interest of LEAs in the presented smart camera; (2) the performance of event detection and of the usability of the visual information in forensic context (Subsection A). In addition, an estimation of the sensor power consumption is reported in Subsection B.

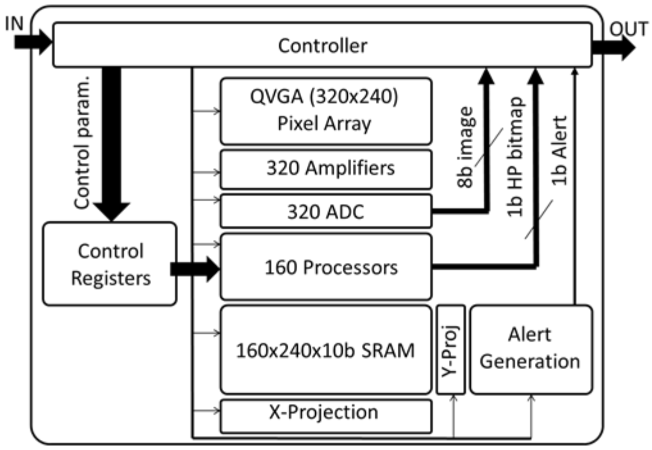

Fig. 5: Block diagram of the low-power vision sensor architecture.

A. Event Detection Performance

As already specified in the Section I, the proposed vision sensor has been developed as a smart tool for supporting the surveillance and forensic activities of LEAs. According to this scenario, we asked LEAs to participate in the evaluation of our technology.

For this task, we used our sensor simulator, reproducing the functionalities of the smart camera, on 30 videos acquired with a commercial VGA monochrome camera [7], where people, involved in the project, simulated possible crimes. To be as much realistic as possible, videos have been captured in different places and under different illuminant conditions.

For each video, the Sensor Emulator computed frame-by-frame the hot pixel map and the hot pixel binary projections, and marked the frames where an alert is generated. Precisely, for each benchmark video, the Sensor Emulator produced a new video, where:

- the left part displays the original input video;
- the right part shows the same video, where, frame-by-frame, the hot pixels (if any) and a maximum bounding box containing them are marked in pink and blue color respectively;
- the bottom banner becomes green as soon an alarm is generated.

The output generated by the Sensor Emulator is shown in Fig. *6*.

The questionnaire consists of six questions:

- Q1: How much such an alert system may help a surveillance officer?
- Q2: How is the video quality, i.e. how evident are the details and the actions depicted in?
- Q3: How accurate is the alert generation indicated by the green banner?
- Q4: In the frames marked by the Sensor Emulator, how accurate is the bounding box delimiting the region of interest, i.e. the *hot-pixels*?
- Q5: In some cases, during an action depicted in the video, some frames are “missed”, i.e. no *hot-pixels* are detected in, so that the green banner is blinking. How much this could adversely affect the intervention of the surveillance officers?
- Q6: Do you think that the video sequence with the event marked by the Sensor Emulator could provide good evidence of a possible crime?

Q1 aims at measuring the interest of LEAs in such a system as a tool for supporting and facilitating their activities. Q2 aims at measuring the quality of the visual information provided by the QVGA format. Q3 aims at measuring the accuracy of the event detection: the participants have been requested to pay attention to possible missed alarms and/or false positives. Q4 aims at evaluating how well the sensor extracts the image region containing all the events detected in the image: this operation could help the surveillance officers to focus their attention on a specific part of the image. Q5 raised up by the analysis of the sensor behaviour: in some cases, we observed that, despite the sensor generates correctly an alarm when an event occurs, the detection of the complete action was interrupted, i.e. 1 or 2 frames were not highlighted as a part of the event (i.e. no *hot-pixels* were detected). Therefore, we asked LEAs to judge the scene understanding from the frames saved by the sensor, i.e. from the frames were the sensor recognized an event. Q6 aims at evaluating the information provided by the sensor as evidences of a crime to be presented in a court.

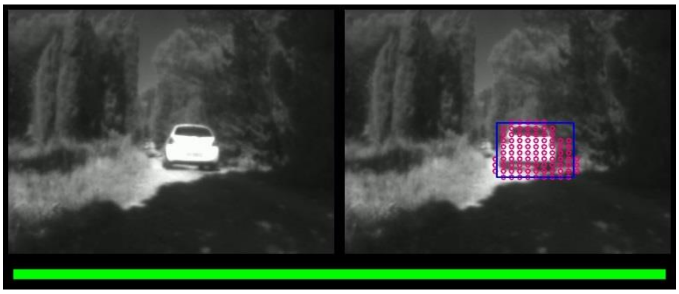

Fig. 6: An example of a video output by the Sensor Emulator and used for evaluating the sensor performance. On left: a frame of the input video. On right: the same frame with hot pixels (in pink) and their maximum bounding box (in blue). On bottom: the banner is green, indicating that an event has been detected.

The questionnaire has been filled in by 15 surveillance officers of the Local Policia of Valencia (Spain) and of the Policia

Judiciaria of Lisbona (Portugal). The participants were asked to give a vote from 1 (unsatisfactory) to 10 (very good) to Q1, Q2, Q3, Q4 and Q6, while a vote from 1 (no matter) to 10 (bad, useless evidence) to Q5.

The results are summarized by the plot in Fig. 7 that reports the average votes given by the participants. These results show a high interest of LEAs in such a sensor and a high satisfaction about its event detection performance.

B. Power Consumption

As shown in TABLE 1, the sensor power consumption has been estimated for a frame rate of 15 fps, 3.3V for the analog part and 1.2V for digital core and IOs. Three operating modes are considered for different duty cycles:

- **no event (NE).** The sensor does not deliver any information off-chip;
- ***event detected* (EB)**. The sensor delivers the hot-pixel bitmap to the processor;
- **event detected (EBG).** The sensor delivers both hot-pixel bitmap together with the grey-level image to the processor.

It is worth to notice that the duty-cycle associated to each functional block depends on the specific application.

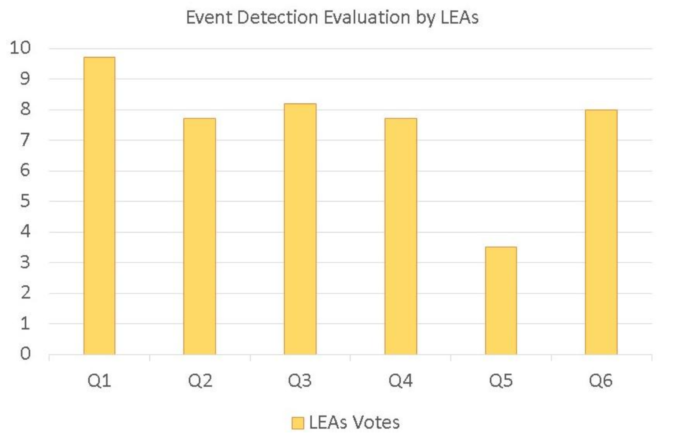

Fig. 7 Summary of the results from the questionnaire evaluating the event detection performance. The votes reported here are averaged over the number of participants to the questionnaire.

The minimum power consumption of the sensor is in Mode 1, when no event is detected, thus no data are delivered by the camera. The power consumption starts increasing as soon as the sensor starts communicating with the processor, dispatching only the hot-pixel bitmap (Mode 2) or the entire grayscale image (Mode 3). However, if we consider that in a typical application the duty cycle for Mode 2 might be 3% while that one of Mode 5 is 2%; under these assumptions the estimated average power consumption of the sensor will be about 860μW. This means that, in a typical surveillance application, powered with 2 AA-1.5V batteries (2850 mAh), the presented smart camera can operate for about 11 months. In this estimation, we didn't take into account the power consumption of the system, which will be relevant according with the number of detected alerts in the operating period.

VI. CONCLUSIONS

In this work, we described the basic architecture of a low power smart camera for battery-powered long-lasting operation, which is tailored to event detection as a potential tool for crime fighting. The sensor can be used to monitor remote areas, with infrastructures, that might be chosen for illegal activities (e.g. traffic of weapons, cigarettes, drugs). The simulated event detection performance of the camera have been evaluated by a group of LEAs employers, that judged the accuracy of event detection and the usability of the visual information provided by the camera as crime evidences in a court. The LEAs responses denoted a high interest in such a camera and a high satisfaction for its performance.

The first version of the presented vision sensor, with reduced functionalities, is going to be tested in real application scenarios. The final camera will be embedded into a wireless node to be part of a network of vision systems capable of covering large areas to be monitored.

TABLE 1: Chip Estimated Power Consumption at 15fps

Chip Block	**Mode 1 No Event**	**Mode 2 Hot Pixels**	**Mode 3 Gray Level**
Pixel Amplifiers	645 μW	645 μW	1290 μW
DAC	93.5 μW	93.5 μW	93.5 μW
Processor Array	112 μW	112 μW	112 μW
IO Pads	0	12 μW	96 μW
TOTAL	**850.5 μW**	**862.5 μW**	**1591.5 μW**
Duty Cycle	1	0.1	0.01

ACKNOWLEDGMENT

The authors would like to express their gratitude to G. Lerma and P.J. Moreno and to all the employees of the Valencia Local Police (ES), to the Policia Judiciaria of Portugal (PT), Almaviva (I) and to the researchers of CERTH (EL) contributing to the evaluation questionnaire. The authors also thank EMZA (IL) for its support on the event-detection algorithm. This work was funded by the EU H2020-FCT-2014 FORENSOR (FOREnsic evidence gathering autonomous seNSOR) Project under Grant n.653355.

[1] A. N. Belbachir (Ed.), Smart cameras, Heidelberg: Springer, 2010.

[2] M. Rusci, D. Rossi, M. Lecca, Gottardi, E. Farella and L. and Benini, "An event-driven ultra-low-power smart visual sensor.," *IEEE Sensors Journal*, vol. 16, no. 13, pp. 5344-5353, 2016.

[3] M. Lecca, M. Gottardi, E. Farella and B. and Milosevic, "Always-on low-power optical system for skin-based touchless machine control," *J. Opt. Soc. Am. A* , vol. 33, no. 6, pp. 1015-1024, 2016.

[4] Z. Smilansky, "Miniature autonomous agents for scene interpretation". US Patent 7,489,802, February 2009.

[5] J. JC, J. CR and M. SR, "Background subtraction and shadow detection in grayscale video sequences," in *Computer Graphics and Image Processing*, Brazil, 2005.

[6] M. L and P. A., "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transactions on Image Processing,* vol. 17, no. 7, pp. 116811-77, 2008.

[7] S.-C. SC and K. C., "Robust techniques for background subtraction in urban traffic video," *Electronic Imaging,* pp. 881-892, 2004.

[8] B. S, H. B and H. G, "Evaluation of background subtraction techniques for video surveillance," in *Computer Vision and Pattern Recognition* , 2011.

[9] H. W, T. T, W. L and M. S., "A survey on visual surveillance of object motion and behaviors," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews),* vol. 343, pp. 334-352, 2004.

[10] V. D. M. V. Barnich O, "A universal background subtraction algorithm for video sequences," *EEE Transactions on Image processing,* vol. 20, no. 6, pp. 1709-1724, 2011.

[11] N. Cottini, M. Gottardi, N. Massari, R. Passerone and Z. Smilansky, "A 33 mu W 64 x 64 Pixel Vision Sensor Embedding Robust Dynamic Background Subtraction for Event Detection and Scene Interpretation.," *IEEE Journal of Solid-State Circuits,* vol. 48, no. 3, pp. 850-863, 2013.

[12] A. Berkovich, M. Lecca, L. Gasparini, p. A. Abshire and M. and Gottardi, "A 30 µw 30 fps 110× 110 pixels vision sensor embedding local binary patterns," *IEEE Journal of Solid-State Circuits,* vol. 50, no. 9, 2015.

[13] https://www.ptgrey.com/firefly-mv-usb2-cameras, Point Grey Firefly MV.