A CMOS Vision Sensor For Background Subtraction

D. García-Lesta, P. López, V.M. Brea, D. Cabello
Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS)
Universidade de Santiago de Compostela
Santiago de Compostela, Spain
Email: daniel.garcia.lesta@usc.es

***Abstract*—Background subtraction is one of the first steps in many video processing algorithms. Thus, a real-time processing with low power consumption is convenient for different applications where power hungry devices with high computational capabilities can not be deployed. This work presents the design of a 24×56 pixel proof-of-concept 0.18 μm standard CMOS vision sensor chip implementing the foreground detection algorithm Hardware Oriented Pixel Based Adaptive Segmenter (HO-PBAS) on the focal plane. Simulation results show a maximum processing speed of 2000 fps with a figure of merit of 1.3 μW/pixel at 60 fps and a pixel pitch of 47 μm in a four pixels per processing element configuration.**

***Index Terms*—CMOS vision sensor, focal plane, foreground detection, HO-PBAS**

I. Introduction

Computer vision algorithms like tracking by detection require to detect the foreground of the image, or equivalently to remove the background [1]. Due to the importance of this step, many different approaches and algorithms have been published in the last decades [2].

Pixel-level foreground detectors are good candidates to be used on embedded platforms as they feature a high level of parallelism. These platforms might be general purpose devices, as GPU-CPU boards or FPGAs, which provide a high level of programmability [3], [4]. If the lowest possible power consumption is pursued, mixed-signal ASICs are usually preferred. These can be general-purpose devices that provide some level of programmability to run the desired algorithms [5]–[7]. However, as they are designed for general applications, complex algorithms that require specific resources can not be fitted in them without important simplifications, which might lead to a loss of performance with respect to the original computer vision algorithm under study. In those cases, specific-purpose ASICs become a feasible option [8], [9]. In that line, reference [10] shows the design of a mixed-signal 64×64 pixels vision chip for background subtraction with a very low power consumption. The vision chip implements a foreground detector based on the difference of the actual frame with respect to a background model formed by a running average of the last frames, which in turn is compared with an adaptive threshold. Even when no visual metrics are given, more recent and elaborated background subtraction algorithms with better performance have arisen. Pixel Based Adapted Segmenter (PBAS) is one of such algorithms [11]. This paper introduces a CMOS vision chip with a modified version of the original PBAS for its mapping on the focal plane without degradation of background performance metrics.

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skodowska-Curie grant agreement No 860370, the Consellería de Cultura, Educación e Ordenación Universitaria (2016-2019, ED431G/08 and ED431C2017/69), the Ministerio de Economía, Industria y Competitividad (TEC2015-66878-C3-3-R and RTI2018-097088-B-C32) and the European Regional Development Fund (ERDF).

II. Foreground detector

The algorithm implemented in this work is a hardware oriented version of the PBAS. This simplification was developed in [12], showing that with linearized equations and less samples of the background model the same performance in grayscale images as with the original algorithm can be achieved.

The segmentation mechanism described in (1) was first introduced in [13]. The strategy consists of counting how many samples of the background model $B_k(x_i)$, formed by previous samples of the pixel, are inside a sphere centered at the input pixel value $I(x_i)$ and with a certain radius $R(x_i)$. If this number is smaller than the parameter $\#_{min}$ the pixel will be considered foreground and background otherwise.

$$S(x_i) = \begin{cases} 1, & \#\{dist(I(x_i), B_k(x_i)) < R\} < \#_{min} \\ 0, & \text{else} \end{cases} \quad (1)$$

The Hardware Oriented PBAS (HO-PBAS) uses a feedback scheme similar to that of the original PBAS to tune the algorithm parameters responsible for the background model update probability $p(x_i)$ and the segmentation sphere radius $R(x_i)$. Both parameters depend on how often a pixel is segmented as background or foreground and also on the dynamics of its background model. To measure the background dynamics the HO-PAS takes the difference between the maximum and the minimum samples weighted by a fixed constant β as:

$$d(x_i) = \beta \cdot [max_k(B_k(x_i)) - min_k(B_k(x_i))] \quad (2)$$

The background dynamics estimator $d(x_i)$ is then used to update the probability parameter:

$$p(x_i) = \begin{cases} p_{min}, & \text{if } S(x_i) = 1 \\ p(x_i) + [1 - d(x_i)] \cdot p_{inc}, & \text{else} \end{cases} \quad (3)$$

where p_{min} and p_{inc} are fixed parameters. Also, $d(x_i)$ is required to calculate the radius $R(x_i)$ of the sphere in the segmentation process, multiplying it by the fixed parameter R_{scale}:

$$R(x_i) = d(x_i) \cdot R_{scale} \quad (4)$$

When the foreground detection is done, the background model might be updated through two different mechanisms:

- Self-update: if the pixel is segmented as background, a randomly chosen sample might be updated based on the pixel dependant parameter $p(x_i)$.
- Diffusion: same as the previous method, if the pixel is segmented as background it might induce a neighbor to update its background model, even if it was segmented as foreground. With that, the problem of static foreground objects is attenuated.

III. In-Pixel Circuits

This work implements a proof-of-concept chip with an array of 24×56 pixels. The complexity of the algorithm leads to a distribution of in-pixel circuitry and circuits shared by a group of pixels, making up the so-called Processing Element (PE). Also, some operations are performed many times across the pipeline of the algorithm. Thus, it is convenient to reuse analog primitives common to many functions along the datapath such as the arithmetic unit shown in Fig. 1. This circuit is a switched-capacitor differential amplifier based on a high gain inverting cascode amplifier. After a full cycle of the non-overlapped inverted clock signals *phi1* and *phi2* the output voltage will be:

$$V_{out} = V_3 + \frac{C_1}{C_2}(V_1 - V_2) \quad (5)$$

Where C_1 and C_2 are the capacitance of the Metal-Insulator-Metal (MIM) capacitors and V_1, V_2 and V_3 the input voltages. Fig. 2 shows a simulation for this circuit where a high linearity and robustness against Monte Carlo simulation can be seen.

Image capturing is carried out by a standard 3 transistors active pixel sensor (3T-APS) with a correlated double sampling (CDS) implemented with the arithmetic unit in Fig. 1. After the integration time, the output of the CDS will be a voltage in a range suitable for the subsequent processing circuits. Also, to reduce the integration time, the CDS block adds a voltage gain naturally implemented through the relationship between capacitors C_1 and C_2 in (5).

A low-pass filter is applied to the image to reduce the noise, improving the algorithm performance [12]. This is done by sharing the charge stored in capacitor C, corresponding to the output voltage of the CDS, through the exchange capacitor C_E in the 4-neighborhood single Euler network shown in Fig. 3 [9]. With this architecture, the parameter σ of the Gaussian distribution can be controlled by the number of clock cycles, n, of the two non-overlapping signals *phi1* and *phi2* as:

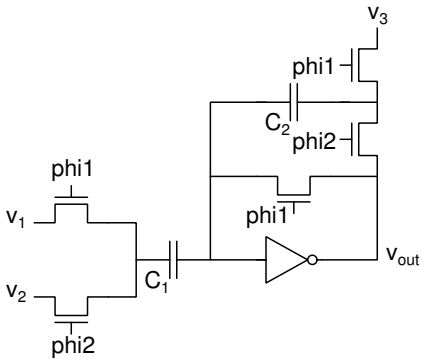

Fig. 1. Schematic of the arithmetic unit circuit used as the analog primitive function in our design.

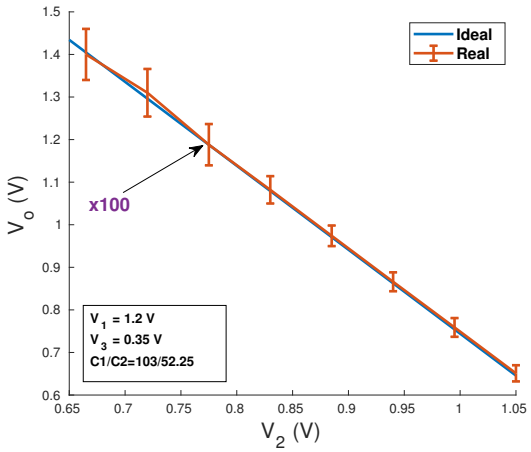

Fig. 2. Arithmetic unit simulation. Vertical bars indicate variation under Monte Carlo simulations, scaled up 100 times.

$$\sigma = \sqrt{\frac{2nC_E}{C}} \quad (6)$$

Once the image is captured and low-pass filtered, it will need to be compared with the background model that is stored per-pixel. To ensure adequate retention times for these values an in-depth analysis of the Analog Random Access Memories (ARAM) architecture was developed in [12], assessing how the performance of HO-PBAS is affected by circuit non-idealities. The main conclusion extracted from this work is that every memory cell needs its own output buffer, which we implement with source followers as in Fig. 4.

The next step in the algorithm datapath is to calculate $d(x_i)$ as in (2) with the circuit in Fig. 5. This circuit extracts the maximum and minimum values from the background model and performs the difference weighted by the fixed parameter β. Once this value is obtained it can be used in (3) and (4) implemented also with arithmetic units as that in Fig. 1.

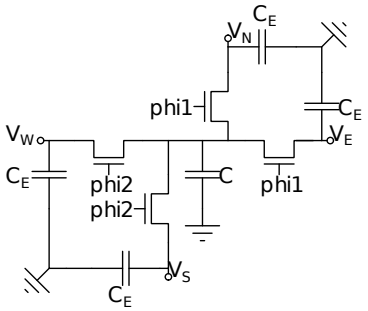

Fig. 3. 4 neighborhood single Euler network for Gaussian filtering.

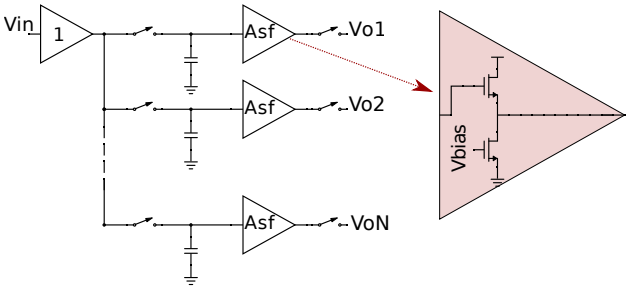

Fig. 4. Analog memories of the chip, with an input buffer implemented with an OA, and as many source followers as units of memory; N in this example.

The last circuit required is the one responsible for the segmentation decision. This circuit needs to get each background model sample, take the absolute difference with respect to the input value $I(x_i)$, compare it with the radius of the sphere and count how many of them are inside. This is implemented in the circuit of Fig. 6. The reason why the counter is only two bits is because the algorithm was optimized through computer vision simulations against the benchmark changedetection resulting in a $\#_{min}$ parameter of 2 [14]. Thus, when the counter reaches this value the pixel can be considered background, regardless of whether the number of samples inside the sphere is bigger than or equal to $\#_{min}$. Thus, when the counter reaches a value of two, b1 is set to high and that triggers the SR latch, which will hold the segmentation result until the next iteration, when it will be reset to a value of $S(x_i)$=1.

Fig. 7 shows results for a segmenter simulation, where two different input values, $I(x_i)$, are simulated with the same background model. In the first part of the simulation, during the first 100 μs, a value of $I(x_i)$ whose difference with respect to the background model samples is bigger than $R(x_i)$ is tested. As the comparator output is always zero, the counter does not trigger the latch and the output remains high (between 80 μs and 100 μs), indicating that the pixel is segmented as foreground. In the second part of the test the input value $I(x_i)$ is modified to a value similar to that of 4 samples of the background model. Thus, when it is compared with them, as the difference is smaller than $R(x_i)$ the comparator produces 4 pulses, and it can be seen that at the second one the counter reaches $\#_{min}$, triggering the SR latch and giving a

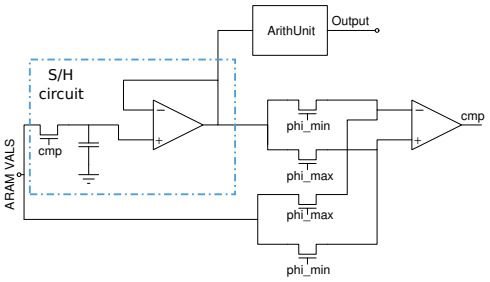

Fig. 5. Background dynamic estimator. The background model values are introduced into the ARAM VALS node one by one and depending on which one of signals phi_max or phi_min is set to high, the maximum or the minimum is stored in the SH circuit at the end of the cycle.

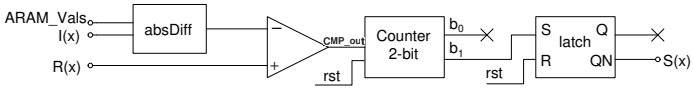

Fig. 6. Segmenter circuit to decide whether or not a pixel is foreground on our chip.

low output voltage (between 180 μs and 200 μs), meaning that the segmentation process has finished with the pixel classified as background.

IV. Chip Architecture

All circuits described in Section III are meant to be placed near the sensor. Nevertheless, the complexity of the HO-PBAS leads to an architecture where all the circuits that can be shared by a group of pixels are placed in a common processing unit. Sharing these circuits reduces parallelism while a complete in-pixel datapath would lead to a very large pixel pitch. A compromise value of a shared processing unit for every group of four pixels has been found, forming all of them the PE. In our solution, the circuits that compute (1),(2),(3), and (4) are those shared by four pixels, as shown in the central block of Fig. 8. Sharing these circuits leads to a careful layout design in order to achieve a homogeneous photodiode array pattern to reduce image distortion. The pixel on the other hand, includes the circuits that can not be shared, such as the ARAM, the local logic needed to implement the background update tasks, the Gaussian blurring circuit, the output to ADC selector (imaged read, image after Gaussian filtering, contents of the analog memory or the $p(x_i)$ value of each pixel) or the frame buffer to hold the captured image until it is read.

Fig. 9 shows the full layout of the 24×56 pixels proof-of-concept 0.18 μm standard CMOS 1.6×3.2 mm^2 chip. The readout of the chip is as follows: first, after the integration time, a row is selected and their captured voltages are connected to a column-level 8 bit single slope analog-to-digital converter (ADC). The result of the conversion is stored in an 8×56 bit row buffer that is accessed through the column decoder. This result is read out while the next row is being converted. When this process is finished, the same row and column decoders repeat the array scan, connecting the result of the segmentation, stored in a latch at each pixel, to the outside.

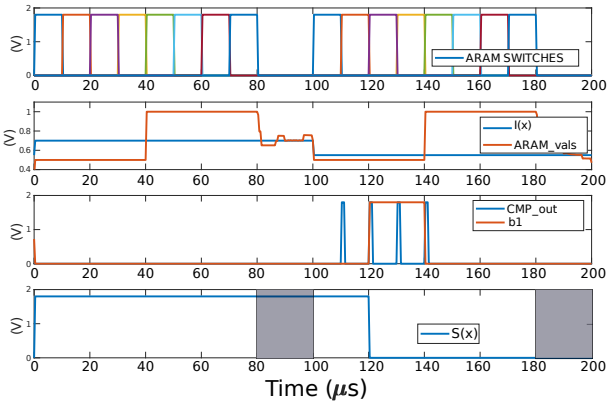

Fig. 7. Segmenter simulation with two different input values (700 mV and 550 mV) against the same background model ($B_k(x_i) = \{0.5, 0.5, .5, 0.5, 1, 1, 1, 1\}$ V). $S(x_i)$ is hold in the grey zones to be read by subsequent circuits.

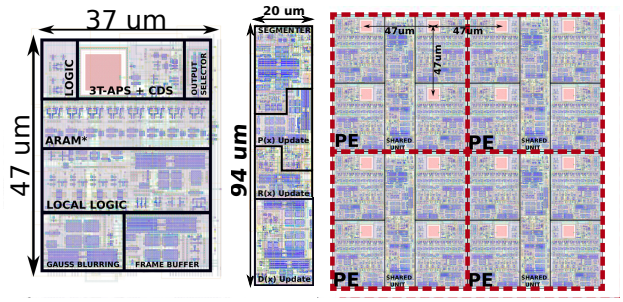

Fig. 8. Chip architecture. From left to right: pixel, shared processing unit and four PEs (formed by four pixels and a shared processing unit each one) that make part of the chip core.

The HO-PBAS needs a random source of analog and digital values, which are obtained from a Random Number Generator based on chaotic maps [15]. The control signals generation is carried out by a global module, in a single instruction multiple data (SIMD) architecture. This block was designed with a hardware description language and generated with digital synthesis tools. This control block is provided with a two bit clocked input bus that reads instructions from the outside. These instructions are intended to manage the integration time, the number of cycles of the Gaussian filtering and the signal that is connected to the ADC. Among the global control, each pixel features some local logic circuits for individual decision making (such as updating or not the background model based on their $p(x_i)$ value or due to the diffusion process). The combination of the local and global logic circuits generates the control signals in a synchronized manner to implement the flowchart shown in Fig. 10, where the sequential processing of each one of the four pixels per PE can be seen.

V. Performance Analysis

As part of the processing circuits is shared by a group of four pixels, in order to make a fair comparison with other works we will consider the metrics of the pixel as the fourth part of the total from the PE. Thus, from simulation results we

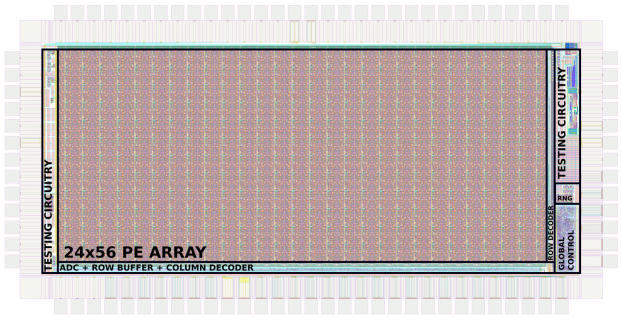

Fig. 9. 24×56 pixel array proof-of-concept chip.

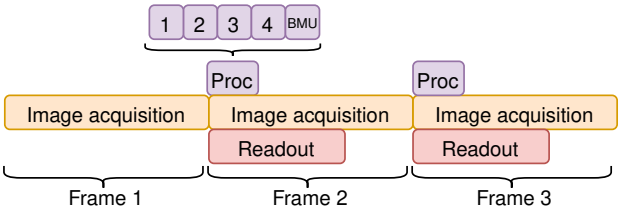

Fig. 10. Algorithm implementation flowchart (BMU: Background model update, Proc: Image processing).

extract a power consumption of 1.3 μW/pixel, or equivalently 21.7 nJ/pixel/frame at a framerate of 60 fps. This power consumption can be considered low power if compared with digital-based systems and in the order of other state-of-the-art vision chips [16]. However, it is considerable bigger than [10], where a total power consumption of 33 μW for a 64×64 pixels vision sensor running at 13 fps is reported. In that case a background subtraction algorithm with worse vision metrics than HO-PBAS is implemented with only 45 transistors per pixel. In our case a more complex algorithm is implemented that requires an equivalent number of 280 transistors and 17 MIM capacitors per pixel. Another consequence of the high pixel complexity is that the pixel pitch is 47 μm, with an 8×8 μm^2 photodiode. However, chip simulations show that frame rates up to 2000 fps can be executed in our design, with the full performance of the HO-PBAS with just an increase in the power consumption by a factor of five.

VI. Conclusions

This paper describes a CMOS vision sensor that integrates both the acquisition and processing stages of a foreground detector on the focal plane. First, a simplification of a state-of-the-art background subtraction algorithm was presented, reaching the same performance as that of the original algorithm with less computational resources. Then, analog circuits to implement the algorithm were developed. Finally, a custom made SIMD architecture was developed for the integration of that circuitry, designing a per-group of pixels approach to reduce the required area. This design was implemented in a 24×56 pixel proof-of-concept chip with a maximum processing speed of 2000 fps, a figure of merit of 1.3 μW/pixel at 60 fps and a pixel pitch of 47 μm in a four pixel per processing element configuration.

REFERENCES

[1] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Aug 2017, pp. 1–6.

[2] T. Bouwmans, "Traditional and recent approaches in background modeling for foreground detection: An overview," *Computer Science Review*, vol. 11, pp. 31–66, 2014.

[3] B. Blanco-Filgueira, D. García-Lesta, M. Fernández-Sanjurjo, V. M. Brea, and P. López, "Deep Learning-Based Multiple Object Visual Tracking on Embedded System for IoT and Mobile Edge Computing Applications," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5423–5431, June 2019.

[4] T. Kryjak, M. Komorkiewicz, and M. Gorgon, "Real-time hardware–software embedded vision system for ITS smart camera implemented in Zynq SoC," *Journal of Real-Time Image Processing*, pp. 1–37, 2016.

[5] S. J. Carey, D. R. Barr, B. Wang, A. Lopich, and P. Dudek, "Mixed signal SIMD processor array vision chip for real-time image processing," *Analog Integrated Circuits and Signal Processing*, vol. 77, no. 3, pp. 385–399, 2013.

[6] M. Laiho, J. Poikonen, and A. Paasio, "Mipa4k: Mixed-mode cellular processor array," in *Focal-Plane Sensor-Processor Chips*. Springer, 2011, pp. 45–71.

[7] D. Ginhac, J. Dubois, B. Heyrman, and M. Paindavoine, "A high speed programmable focal-plane SIMD vision chip," *Analog Integrated Circuits and Signal Processing*, vol. 65, no. 3, pp. 389–398, 2010.

[8] A. Zarándy, *Focal-Plane Sensor-Processor Chips*, 1st ed. Springer-Verlag New York, 2011.

[9] M. Suárez, V. M. Brea, J. Fernández-Berni, R. Carmona-Galán, D. Cabello, and Á. Rodríguez-Vázquez, "Low-power CMOS vision sensor for gaussian pyramid extraction," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 2, pp. 483–495, 2017.

[10] N. Cottini, M. Gottardi, N. Massari, R. Passerone, and Z. Smilansky, "A 33 uW 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.

[11] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference*. IEEE, 2012, pp. 38–43.

[12] D. García-Lesta, P. López, V. M. Brea, and D. Cabello, "In-pixel analog memories for a pixel-based background subtraction algorithm on CMOS vision sensors," *International Journal of Circuit Theory and Applications*, vol. 46, no. 9, pp. 1631–1647, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cta.2458

[13] O. Barnich and M. Van Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image processing*, vol. 20, no. 6, pp. 1709–1724, 2011.

[14] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, "Changedetection.net: A new change detection benchmark dataset," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference*. IEEE, 2012, pp. 1–8.

[15] J. L. Valtierra, E. Tlelo-Cuautle, and Á. Rodríguez-Vázquez, "A switched-capacitor skew-tent map implementation for random number generation," *International Journal of Circuit Theory and Applications*, vol. 45, no. 2, pp. 305–315, 2017.

[16] Z. Chen, H. Zhu, E. Ren, Z. Liu, K. Jia, L. Luo, X. Zhang, Q. Wei, F. Qiao, X. Liu, and H. Yang, "Processing Near Sensor Architecture in Mixed-Signal Domain With CMOS Image Sensor of Convolutional-Kernel-Readout Method," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 1–12, 2019.