

A 120 μ W vision chip with ROI detection

Arnaud Verdant, Antoine Dupret, Patrick Villard,
Laurent Alacoque
CEA - LETI – MINATEC Campus
F-38054 Grenoble, France

Hervé Mathias, Flavien Delgheier
IEF
Université de Paris sud 11
F-91400 Orsay, France

Abstract— A smart ultra-low power CMOS image sensor comprising an analog programmable processor array is reported. Compact and efficient motion detection algorithms are implemented to process sub-sampled images made of so-called macropixels. Only Regions of Interest (ROI) consisting of macropixels containing moving objects are read out. This drastically reduces power consumption: the 110 \times 240 pixel image sensor fabricated in a 0.35 μ m technology features a power consumption of 120 μ W at 25fps.

I. INTRODUCTION

Achieving ultra-low power consumption is of most importance for battery powered image sensors. In the case of video surveillance applications, this power consumption can be adapted to the scene activity. However, the detection of intrusions implemented on processors associated to video cameras wastes a considerable amount of power. Indeed, every frame of the whole image array is processed, whereas the actual scene is steady, or merely comprises very few and small regions of interest (ROIs). Several publications [1-5] have reported the reduction of power consumption in image sensors (Table I). Most of them rely on event detections based on simple 2-frame differences. However, this approach is not robust enough [6]. Therefore complex algorithms are mandatory to insure a low False Alarm Rates.

In this paper, we present a CMOS image sensor capable of efficiently detecting and tracking ROIs with ultra-low power consumption. The circuit has been optimized to implement various motion detection algorithms, from which derives the locations of ROIs. Those algorithms based on the difference between the actual scene and an estimate of its background, obtained from the temporal filtering of the macropixels' luminance, are among the most compact ones [7].

II. ARCHITECTURE

A. Architecture overview and functioning principle

The image sensor proof of concept is composed of a 240 \times 110 pixel array associated to a Single Instruction Multiple Data (SIMD) vector of 11 general purpose analog programmable processing units (PUs) and to 5 banks of 24 \times 11 analog memory array (ARAM) (Fig. 1).

Having analog programmable processing units prevents using ADCs and so contributes to reduce the power consumption and the silicon area.

Yet, much like digital processors, the PUs feature a high degree of flexibility. Indeed their instruction set allows implementing various complex algorithms (e.g. enhanced versions of $\Sigma\Delta$ or Recursive Average [7]).

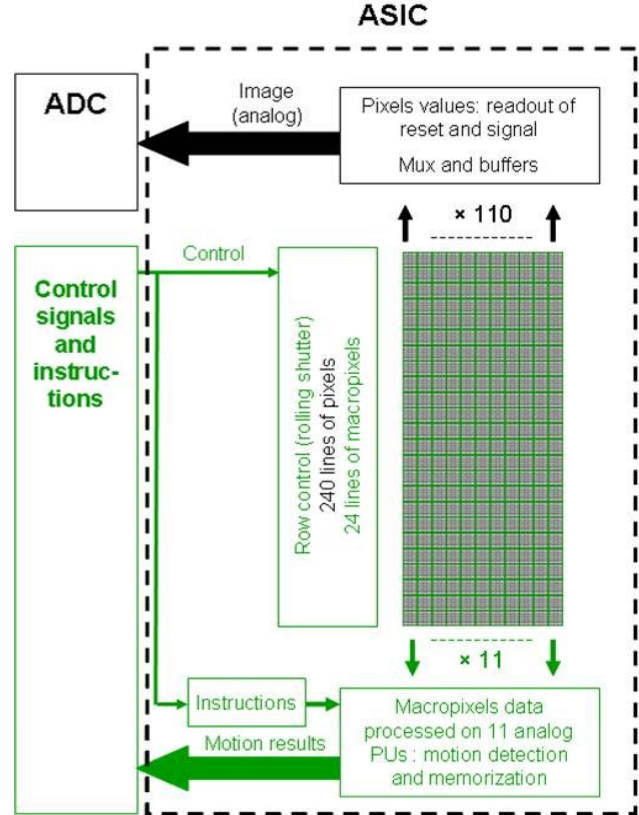


Figure 1. Global ASIC architecture

The computations are performed on the local average of 10 \times 10 pixels generated from 10 \times 10 connected photodiodes spread among the considered macropixels. These averages are acquired in rolling shutter mode, as well as full resolution sub-images (ROIs). By using this sub-sampling scheme, the data throughput towards the processing elements is divided by 100, while keeping the contribution of all the pixel. The results of the computations are used to detect critical temporal activity, considered as moving objects, and to give the location of the regions of interest (ROIs) where moving objects have been located. As shown in Fig 1, only the pixels within the macropixels elected as ROIs are driven outside the sensor by a readout pipeline opposite to the vector of PUs. As long as no motion appears, the high resolution pixels are not read out, which contributes to further reduce the power consumption. The synoptic describing the general sensor behavior is exposed on Fig. 2.

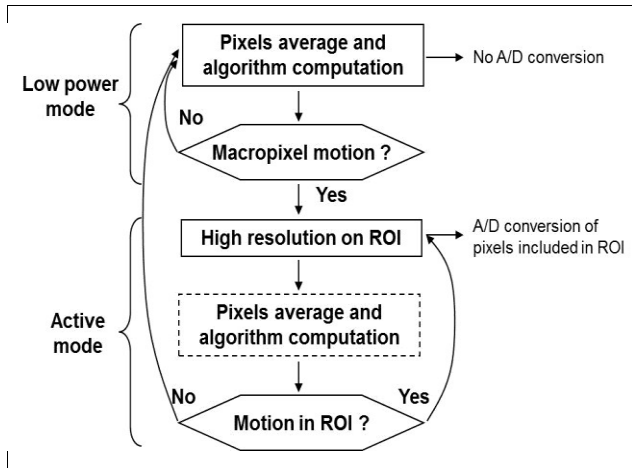


Figure 2. Synoptic describing sensor behaviour for motion detection processing and high resolution pixel readout

B. Main blocks

The photosensor array comprising 3T pixels of $10\mu\text{m}$ pitch with 50% fill factor is depicted on Fig. 3. Both low and high resolution images are acquired in the rolling shutter mode from two interleaved photosensor arrays. The chip comprises 11×24 macropixels. The photodiode of a macropixel is spread among all its constituting pixels except one. Indeed, in one pixel out of 100, the macropixel photodiode is replaced by the three transistors of the 3T architecture to allow the reset and reading of the macro-pixel photodiodes. For each column of pixels, a switch allows selecting a pixel being part of a ROI. These switches are controlled by the signals cmd_i according to motion detection results. The high resolution ROIs are then read out from the pixel array by a standard readout pipeline.

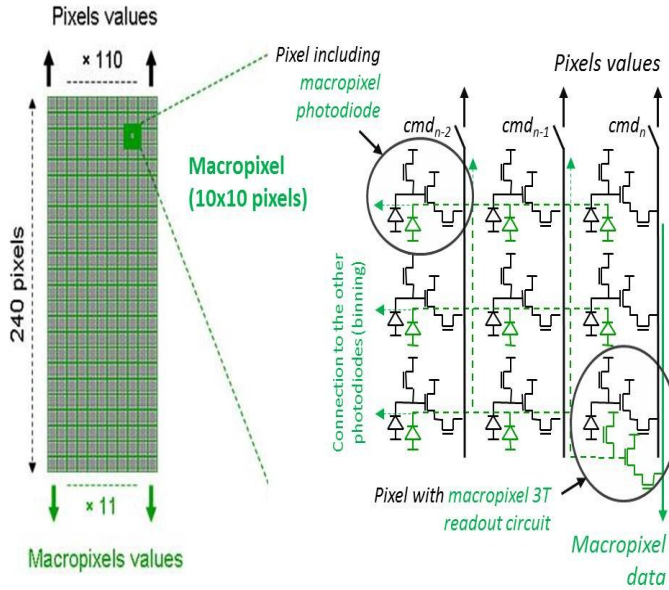


Figure 3. Photosensors array and macropixel architecture (simplified example on a macropixel made of 3×3 pixels)

The PU attached to each of 24 columns of macropixel (Fig. 4) consists of switched capacitor circuits with reduced instruction set (multiply and accumulate (MAC), absolute value, comparison, conditional operation, Boolean operators, read and store). These analog based PU perform computations on analog values according to the instructions they have received. The clock signals, which control the switches, are generated according to the instruction to be performed. The instruction frequency is kept low (i.e. 100 kHz) to lower the power consumption. Yet such a low clock frequency still allows implementing over 150-instruction program at 25fps. This is still important since previous works have shown that the implementation of many compact background estimation algorithms requires less than 50 instructions [7] on such processors. Among the set of 16 instructions, the main ones are: multiply and accumulate (MAC), absolute value, read, store, comparison, conditional operation, Boolean operators.

An Analog RAM (ARAM) is associated to each processing unit. The so-formed ARAM array attached to the PUs contains a total of $24 \times 11 \times 5$ memory points, allowing to store up to 5 data per macropixel, according to the algorithm requirements. For each PU, multiplexers allow to select the rows of macropixel or analog memories to be processed. Using analog memories introduces an error for each storage operation. This error must be kept small to avoid divergence of results in the case of recursive algorithms. This issue has been addressed by the following design: a 500fF capacitor associated to a source follower is used in each memory point.

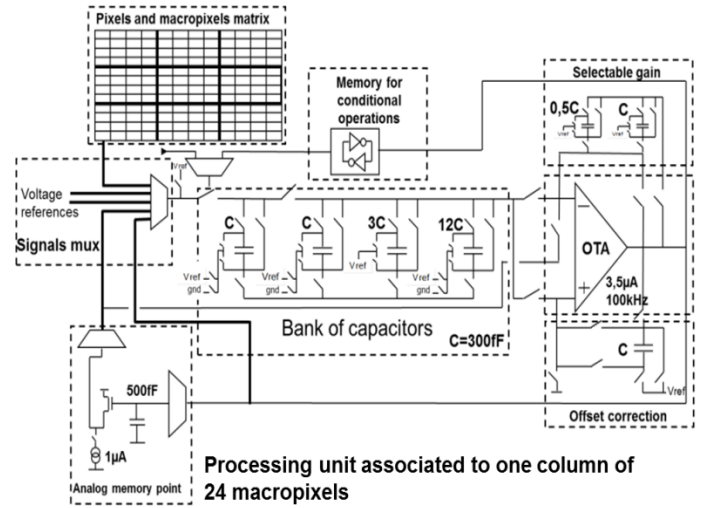


Figure 4. Analog Processing Unit and Analog RAM memory point

A writing mode, placing the memory point in the retroacting path of a closed loop OTA, allows the different non-idealities to be taken into account at each storage operation (Fig. 5). By this way, when reading the capacitor voltage of the memory point, the initial value can be accurately restored: even with the errors introduced by the finite gain of the OTA and by the residue of the offset cancellation, the final error is as low as $250\mu\text{V}$, i.e. less than 0.013% of the dynamic range.

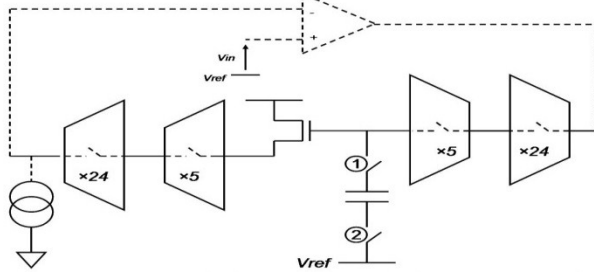


Figure 5. Write operation on one analog memory point (among a total of 24×5 ARAM cells)

The main operation is the MAC that is required to perform filtering operations such as background estimation. In such algorithm, the standard deviation of each macropixel signal is estimated. This standard deviation is then used to estimate a maximum range for background variations. If the value of a considered pixel rises above this estimated range of background variations, we consider that motion occurs.

First of all, background estimation ($RA1_n$) is computed with recursive operation (eq. 1). Temporal variations (Δ_n) are extracted as absolute difference between macropixel signal (S_n) and background estimation (eq. 2). The mean deviation of estimated background variations ($RA2_n$) is then calculated from (Δ_n) (eq. 3). In a fourth step, two variables ($RA3_n$ and $RA4_n$) are computed (eq. 4) and (eq. 5), which allow here to define the estimated range of maximum background variations. Motion is then considered according to (eq. 6).

$$RA1_0 = S_0, RA2_0 = 0, RA3_0 = S_0, RA4_0 = S_0$$

$$RA1_n = RA1_{n-1} - \frac{1}{N} RA1_{n-1} + \frac{1}{N} S_n \quad (\text{eq.1})$$

$$\Delta_n = |RA1_n - S_n| \quad (\text{eq.2})$$

$$RA2_n = RA2_{n-1} - \frac{1}{N} RA2_{n-1} + \frac{1}{N} \Delta_n \quad (\text{eq.3})$$

$$RA3_n = RA3_{n-1} - \frac{1}{N} RA3_{n-1} + \frac{1}{N} (S_n + RA2_n) \quad (\text{eq.4})$$

$$RA4_n = RA4_{n-1} - \frac{1}{N} RA4_{n-1} + \frac{1}{N} (S_n - RA2_n) \quad (\text{eq. 5})$$

$$\text{if } S_n > RA3_n + RA2_n \text{ or } S_n < RA4_n - RA2_n \rightarrow \text{motion} \quad (\text{eq.6})$$

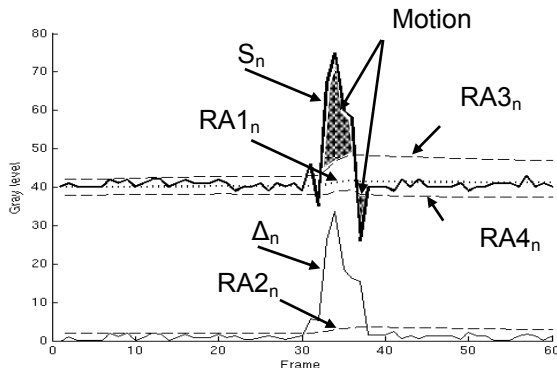


Figure 6. Computation of a macropixel signal. S_n is the macropixel gray level value, with the variables $RA1_n$, $RA2_n$, $RA3_n$, $RA4_n$ and Δ_n as respectively expressed in eq. 1 to eq. 5.

This algorithm so relies on a constant, N , allowing to determine the time constant of recursive averages. However, no additional constant is required to handle sensitivity. Computations of $RA3_n$ and $RA4_n$ allow here to define adaptive thresholding directly from signal variations (Fig. 6).

When a MAC digital instruction word is applied, the switches commute as in Fig.7, which thus implements eq.1. Similarly, the other instructions are based on combinations of switches. The OTA is then used to implement operations that need charge transfer, comparison, etc.

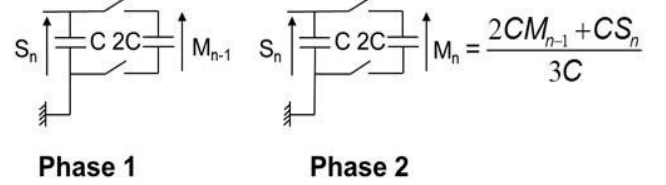


Figure 7. MAC with $N=3$

III. MEASUREMENTS AND IMPLEMENTATION

A. Measurements

Fig. 8 shows the measured result of a recursive average calculation described in the previous section. Measurements have been done on the whole macropixel dynamic range. From the linear regression of this measure, where each point of the curve results from a succession of 15 instructions, an 8 bit processing resolution has been extracted.

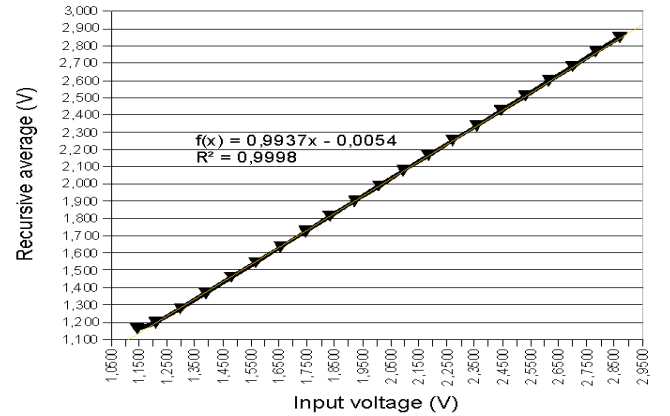


Figure 8. Measurements of computational results of a Recursive average and its corresponding linear regression

B. Implementation

Various compact motion detection algorithms have been implemented, achieving Detection Rate as high as 95% (see [7] for more details about the test bench used to characterize these performances). For test and characterization purposes, the control signals and the Finite State Machine (FSM) that delivers instructions to the PUs have been generated by an external CPLD.

An image acquired with the presented sensor is exposed on Fig. 9. Only the ROIs consisting of the person moving from left to right is read out from the sensor.

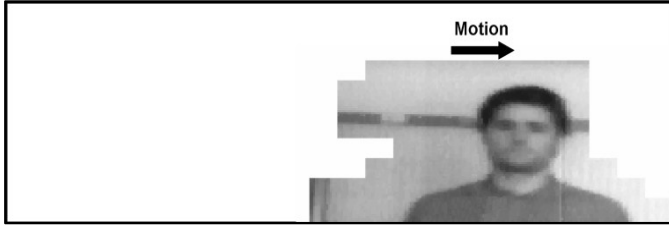


Figure 9. Example of ROI detection

Compared to previous publications (cf. Table I), our chip features the highest fill factor, the lowest power consumption and the lowest energy per pixel. With the recursive average algorithm, the total power consumption of the pixel array, PUs and memory is $120\mu\text{W}$. Using standard batteries of about 1 A.h, the autonomy of the presented sensor reaches one year.

TABLE I. PERFORMANCE AND COMPARISON TO THE STATE OF THE ART

	This work	[5]	[4]	[3]	[2]	[1]
Fps	25	<21	30	50	50	240
Resol.	240×110	32×32	176×144	128×109	64×128	256×256
Program mable	general purpose proc.	no	2 op.	conv. kernels	no	no
Power (mW)	0.12	0.0088	17.6	50	0.1	74,87
Techno. (μm)	0.35	0.35	0.35	0.6	0.35	0.35
Pixel size (μm^2)	10×10	15×29	34×29	10×10	26×26	$8,9 \times 8,9$
Fill-fact.	50%	21%	6.45%	30%	20%	-
Year	2011	2011	2011	2011	2009	2007
FOM (pJ/pix.)	181	409	22359	71674	244	4768

A microphotography of the CMOS $2.4 \times 7.4\text{mm}^2$ chip processed in a $0.35\mu\text{m}$ technology is exposed on Fig. 10.

IV. CONCLUSION

A smart 110×240 pixels image sensor dedicated to ultra-low power application with a general purpose 100k instructions/s SIMD processor array has been presented. Associated to an $11 \times 24 \times 5$ ARAM array, the analog reduced instruction set PUs allow implementing robust and computationally efficient motion detection algorithms. The use of programmable analog processors combines the flexibility of programmable processors without using ADCs. The true programmability and the ultra-low power consumption of $120\mu\text{W}$ are the key differentiators of this smart image sensor. With respect to the number of pixels, this is the best performance reported.

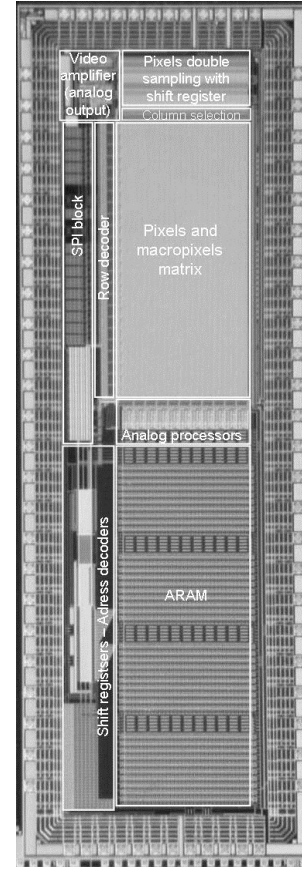


Figure 10. CMOS chip microphotography

REFERENCES

- [1] Jaehyuk Choi; Sang-Wook Han; Seong-Jin Kim; Sun-II Chang; Euisik Yoon; , "A Spatial-Temporal Multi-Resolution CMOS Image Sensor with Adaptive Frame Rates for Moving Objects in the Region-of-Interest," ISSCC 2007., IEEE, vol., no., pp.502-618, 11-15 Feb. 2007
- [2] Gottardi, M.; Massari, N.; Jawed, S.A.; "A 100 μW 128×64 Pixels Contrast-Based Asynchronous Binary Vision Sensor for Sensor Networks Applications," Solid-State Circuits, IEEE Journal of , vol.44, no.5, pp.1582-1592, May 2009.
- [3] Njuguna, R.; Gruev, V., "Low Power Programmable Current Mode Computational Imaging Sensor," Sensors Journal, IEEE , vol.PP, no.99, pp.1, 0, doi: 10.1109/JSEN.2011.2158579.
- [4] Fernández-Berni, J.; Carmona-Galán, R.; Carranza-González, L.; "FLIP-Q: A QCIF Resolution Focal-Plane Array for Low-Power Image Processing," Solid-State Circuits, IEEE Journal of , vol.46, no.3, pp.669-680, March 2011.
- [5] Law, M.K.; Bermak, A.; Shi, C.; , "A Low-Power Energy-Harvesting Logarithmic CMOS Image Sensor With Reconfigurable Resolution Using Two-Level Quantization Scheme," Circuits and Systems II: Express Briefs, IEEE, vol.58, no.2, pp.80-84, Feb. 2011.
- [6] A. Manzanera, J. Richefeu, "A robust and computationally efficient motion detection algorithm based on Sigma-Delta background estimation", ICVGIP '04, Kolkata, India, December 2004.
- [7] Arnaud Verdant, Patrick Villard, Antoine Dupret, and Hervé Mathias, "Three Novell Analog-Domain Algorithms for Motion Detection in Video Surveillance," EURASIP Journal on Image and Video Processing, vol. 2011, Article ID 698914, 13 pages, 2011. doi:10.1155/2011/698914.