

# Low Power Motion Detection with Low Spatial and Temporal Resolution for CMOS Image Sensor

Arnaud Verdant<sup>(1)</sup>, Antoine Dupret<sup>(2)</sup>, Hervé Mathias<sup>(2)</sup>, Patrick Villard<sup>(1)</sup>, Lionel Lacassagne<sup>(2)</sup>

(1) CEA/ LETI, Centre de Grenoble, 17 rue des Martyrs, 38054 Grenoble Cedex 9

(2) Institut d'Electronique Fondamentale, Bâtiment 220, Université de Paris 11, 91405 Orsay Cedex  
arnaud.verdant@cea.fr

## Abstract

*Video surveillance aims at detecting unexpected individuals or objects intrusion. When no motion is observed, common motion detection systems induce huge power consumption, regardless of the scene activity. This paper presents algorithms for low power motion detection, and their possible implementation. The main interest is that they are able to adapt the sensor's behavior according to the scene activity. Relevant motion information can be extracted from images with lowered spatial and temporal resolution, with specific algorithms. By reducing the amount of data to analyze and spatial and temporal redundancy, a drastic reduction of power consumption can be achieved.*

## 1. Introduction

In a context of embedded video surveillance, with steady camera, power consumption reduction constitutes a key point. The tremendous amount of data processed independently of scene activity in common image sensors systems (i.e. image sensors associated to a DSP or FPGA), induce huge waste of power. Hence a promising approach consists in "waking up" the system when an event occurs in the scene. Detecting events is then equivalent to motion detection.

For several years, many studies have focused on motion detection problem. Today's submicron processes enable the pixel level implementation of some processing steps that used to be done by external circuits while keeping reasonable Fill Factor and pixel area. Many architectures for motion detection and tracking have been proposed. Here are a few examples of possible detection methods:

- Optical flow measurement has been explored in [1] and [2]. In these approaches, spatial and temporal gradient are calculated to solve optical flow

equation. Each pixel is so computed, increasing at the same time the amount of data for larger images.

- Frame difference is made in [3]. Each pixel data is coded with mantissa and exponent, allowing wide dynamic range, and comparison between frames is done using exponent bits. Therefore, small amplitude motion is difficult to detect with this method.

In the precedent cases, all pixels are used to perform computations. In order to reduce the amount of data to be processed, and so to lower power dissipation, multiresolution architectures have been developed leading to variable spatial acuity imaging, i.e. Region Of Interest (ROI).

- In [4] and [5], authors present architectures able to perform multiresolution outside the pixel array, by sharing charges on banks of capacitors, but increasing die area.
- In [6] and [7], architectures allowing charge binning between pixels inside the array are described, with additional transistors inside and between pixels.
- Multiresolution is achieved by summing pixels currents in [9].

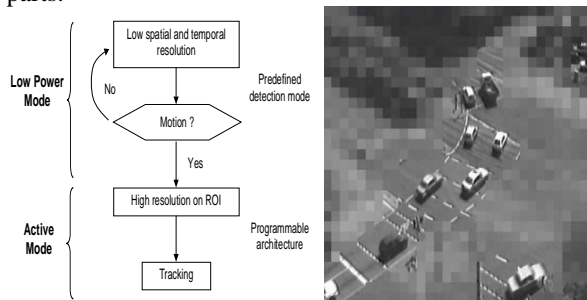
In these approaches with Region Of Interest (ROI), [4] to [9], external chips are necessary to compute addresses of foveated area, increasing power consumption.

- In [10], a computational circuit performs variable resolution by grouping pixels according to their brightness, using quadtree algorithm, but this method requires the whole pixel array to be read.
- Image sub-sampling, in a context of embedded video surveillance, has been explored in [11]. An architecture allowing two operating modes has been developed. Without scene activity, sensor stays in low power mode, and checks the brightness of 1% of all the pixels. If the brightness significantly changes, the sensor switches to active mode and all pixels are read. However, this sub-sampling

technique significantly reduces fill factor [8]. On the contrary, fill factor remains unchanged with macropixels.

Our goal is to develop an imager featuring two operating modes, one being a power saving mode with low spatial and temporal resolution, able to wake up upon a scene event, and the other being normal or active mode, performing tracking on target. To face different situations found in video monitoring, an implementation on a programmable architecture has been chosen (flexible and adaptable architecture).

This paper presents the studies performed concerning the wake up function, i.e. the transition from power saving to active mode. Figure 1 illustrates the two operating modes, with a scene with high resolution on moving cars and low resolution on static parts.



**Figure 1 – Two modes depending on scene activity with high resolution on moving targets and low-resolution for fixed part of the scene**

Part 2 presents our power saving strategy, based on a spatial and temporal data amount reduction, and the considered low resolution configurations. In Part 3, we describe the associated motion detection algorithms that we found potentially interesting. We then (part 4) present the obtained simulation results to determine the best solution. Finally (part 5), we describe the selected implementation architecture and we estimate the power consumption of the selected algorithms.

## 2. Power saving strategy

Within video monitoring tasks, different power dissipation sources may be distinguished:

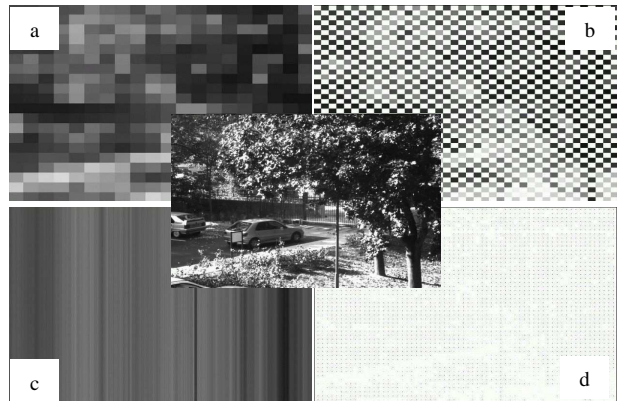
- Image acquisition, image reading and A/D conversion induce a power dissipation that is proportional to the pixel rate.
- Image processing: its related power dissipation not only depends on pixel rate but also on the processed algorithm complexity.

The relative impact of the precedent sources on the global power dissipation depends on the implementation architecture.

The power dissipation lowering may thus be addressed at software and hardware level. For our left-behind video monitoring system, the software-related dissipation saving is obtained by reducing the spatial and/or temporal pixel resolution in order to tremendously reduce the pixel rate in low power mode. The associated motion detection algorithm has to be selected so as to have a number of micro-instructions per pixel as low as possible and yet being efficient enough to detect the aimed events.

Different ways of lowering the spatial resolution have been studied. They are either based on published works or new proposals. Figure 2 shows the considered configurations for a given image. The processed images are in each case split into a number of blocks corresponding to the desired low resolution. 4 different types of blocks have been considered:

- Macropixels (Fig.2 a): blocks containing the spatial average of their constituting pixels;
- Decimated Macropixels (Fig.2 b): a subset of macropixels are processed;
- Uniform stripes (Fig.2 c), containing the average of each column of pixels.
- Decimated pixels (Fig.2 d): blocks where only the central pixel is processed;



**Figure 2 – Original image and corresponding low-resolution images obtained with the different methods considered**

For all the considered configurations, except decimated pixels, all the pixels of a given active block remain active. A spatial average is performed for each block at no extra power consumption by connecting the pixel capacitors in parallel. Only the spatial average (one value per block) is then processed to perform motion detection. These three new configurations have

thus a slightly higher power consumption than the decimated pixels solution, but with far better results in terms of efficiency, as will be seen in part 4.

### 3. Studied Low Power motion detection algorithms

We chose to focus on algorithms aiming at wakening Regions Of Interest (ROI) when movement is detected. They indeed allow limiting the power dissipation during the active mode and easing the operator work.

The considered algorithms are based on Recursive Average (RA), presented in [12], with two modified versions with improved robustness (Sigma Delta and Recursive Average with estimator) at the cost of higher complexity and thus additional power dissipation.

In order to estimate the associated power dissipation, each tested algorithm is described so that the number of necessary micro-instructions may be determined. The following notations have been used: the  $n$  index represents the frame number, the current gray level value for the considered block is named  $S_n$  and a motion estimator is computed and named  $\Delta_n$ . If this estimator becomes larger than a predefined threshold, which depends on the kind of event to be detected, the corresponding and some neighboring blocks are switched to high resolution (all their pixels are processed).

- *Recursive average (RA)*

The principle of recursive average is to estimate background ( $Mrec_n$  (1)) for each data and to compare it to the current value.  $N$  is a fixed coefficient that is chosen with respect to the frame rate and typical target speed:

$$Mrec_n = Mrec_{n-1} - \frac{1}{N}Mrec_{n-1} + \frac{1}{N}S_n \quad (1)$$

$$\Delta_n = |Mrec_n - S_n|$$

- *Modified Sigma Delta (SG2)*

The  $\Sigma\Delta$  algorithm allows estimating background with elementary increment and decrement. Two estimators,  $Msg$  and  $V$ , representing temporal activity and standard deviation of each data, are so generated:

$$\Delta_n = Msg_{n-1} - S_n$$

$$\text{if } \Delta_n > 0 \rightarrow Msg_{n-1} = Msg_{n-1} - 1$$

$$\text{if } \Delta_n < 0 \rightarrow Msg_{n-1} = Msg_{n-1} + 1$$

$$\text{if } V_n < N|\Delta_n| \rightarrow V_n = V_{n-1} + 1$$

$$\text{if } V_n > |\Delta_n| \rightarrow V_n = |\Delta_n| + 1$$

$$\text{if } |\Delta_n| > NV_n \rightarrow \text{motion}$$

As shown above, the threshold on  $\Delta_n$  for motion detection is not fixed here and depends on  $N$ .

- *Recursive average with estimator(RA+E)*

To improve robustness, we have developed this original algorithm based on Recursive Average combined with an estimator, like the one used in SG2 algorithm, which induces more stability with respect to noise. The two combined estimators indeed allow pass band filtering instead of only low-pass filtering in the case of Recursive Average.

$$Mrec_n = Mrec_{n-1} - \frac{1}{N}Mrec_{n-1} + \frac{1}{N}S_n$$

$$\delta_n = Msg_n - S_n$$

$$\text{if } \delta_n > 0 \rightarrow Msg_n = Msg_{n-1} - 1$$

$$\text{if } \delta_n < 0 \rightarrow Msg_n = Msg_{n-1} + 1$$

$$\Delta_n = |Mrec_n - Msg_n|$$

### 4. Results

Simulations have been performed for the considered algorithms with each kind of proposed low-resolution configurations. For these simulations, a single image sequence, that is representative of the aimed applications, has been used. It takes place in an outdoor area where cars are riding a street behind barriers and inside a car park. A tree with rustling foliage is also present and has been included in false alarm parts of the scene. The original resolution of the sequence is 480x640.

**Table 1 – Percentage of processed amount of data with respect to original sequence resolution for tested low resolutions methods**

	1x640	12x16	20x20	40x40	48x64
<b>Macropixels</b>	-	0.06%	0.13%	0.52%	1%
<b>Macropixels Decimation by 2</b>	-	0.03%	0.065%	0.26%	0.5%
<b>Columns</b>	0.21%	-	-	-	-
<b>Pixel Decim.</b>	-	0.06%	0.13%	0.52%	1%

Several levels of low resolution have been tested, as shown on Table 1, which indicates, for each configuration, the percent of data amount processed with respect to the original sequence. Since, for a given algorithm, the power consumption is directly linked to the amount of data processed, this gives a good idea of how much power will be saved with each considered configuration.

For example, in the case of 12x16 low-resolution obtained using macropixels decimation by 2, (12x16)/2

data are processed, which represents 0.03% of data with respect to full resolution. A data amount reduction of 99.97% is so achieved. A comparable power reduction can be expected

The impact of low resolution on efficiency is less straightforward since spatial filtering is performed by the resolution lowering. The best compromise between power saving and efficiency depends on the considered algorithm.

In order to have a better efficiency in motion detection, a ring of fixed size around the processed block has been defined. If a block becomes active, all the blocks encompassed by the ring also switch to active mode. Two ring sizes have been tested: 3x3 and 5x5 blocks. In the case of stripes configurations, a band of adjacent stripes has been used instead.

During the simulation, each time a block is activated, the corresponding information is stored in a vector. At the end of the simulation, this vector is compared to a reference structure indicating where and when motion should be detected. The number of false alarms (blocks which crossed threshold in areas where there is no motion) and of not detected events (blocks which should have crossed threshold in areas where there is motion) can thus be counted. The algorithm latency can also be determined in this manner.

The efficiency of the different tested configurations can then be estimated by subtracting the number of false alarms to the number of rightly activated blocks. A result of 100% corresponds to the detection of all the desired movements in the sequence and no false alarm. For these evaluations, frame rate was 20fps.

A comparison of the obtained results has confirmed that, depending on the considered algorithm, the chosen low resolution configuration has not always the same impact on the efficiency. The best results obtained for the different algorithms are shown in Table 2, which clearly stresses the efficiency of RA+E algorithm with Macropixels configuration. For the Recursive Average algorithm (RA), the results obtained with the decimated pixels and the macropixels configurations have been added in order to show the improvement brought by the proposed low resolution configurations.

**Table 2 – Results of best algorithms with best configurations (M : Macropixels, DM : Decimated Macropixels, D: Decimated Pixels)**

Algorithm	3x3 Ring				Mean value
	12x16	20x20	40x40	48x64	
<b>RA (DM)</b>	<b>95.76</b>	85.38	87.20	76.60	<b>86.2</b>
<b>RA (M)</b>	88.07	80.70	81.98	74.68	81.4
<b>SG2 (M)</b>	88.10	81.46	78.95	71.06	79.9
<b>RA+E (M)</b>	85.53	81.72	82.04	77.58	81.7
<b>RA (D)</b>	29.62	31.13	47.78	45.53	38.6

With a ring of 3x3 blocks, RA+E globally produced the best results with an average evaluation of 86.2%. The best mark obtained by this algorithm is with a 12x16 resolution and Macropixels configuration. In this case, 100% of events are detected with only 4.2% of false alarms.

Increasing rings sizes has globally brought worse results since larger areas are then switched to high resolution. As a consequence, the number of false alarms increases.

The Decimated Macropixels configuration is more efficient with the RA algorithm. The choice of the mesh size is however very important: it must be smaller than the to-be detected objects, in order to have enough reference points on the targets, but it must also be larger than the sources of noise (e.g.: the leaves in the trees for our sequence) in order to reduce the number of false alarms.

Since more stability is achieved with our new RA+E algorithm, the Macropixels configuration is preferable in this case in order to maximize the number of reference points to switch in high resolution on targets. Detection is here excellent with few false alarms since noisy parts (foliage) are efficiently filtered.

Concerning the modified SG2 algorithm, Macropixels configuration is also required but results are less satisfactory because of a less efficient detection.

Using larger resolutions lowers the efficiency of the spatial filtering performed de facto by the low resolution configurations. The macropixels are composed of fewer pixels and the corresponding reference points are less stable. Most of the time, using larger resolutions thus gives bad results, with an increased number of false alarms. The trend is the opposite in the case of decimated pixels.

With the Uniform Stripes configuration associated to the RA algorithm for example, few false alarms are generated (9.41%) and detection is quite performing (89.45%). The main drawback of this configuration is that, when switching to high resolution, a large part of

the image becomes unnecessarily active, thus inducing unnecessary power dissipation. Furthermore, motion detection is only efficient with targets moving perpendicularly to the columns. The Uniform Stripes configuration is also much harder to handle in terms of threshold determination since a given column covers a large part of the image and may contain static parts and noisy parts (foliage in our case). In this case, the corresponding threshold has to be higher to get a reasonable number of false alarms, which reduces the sensitivity for the static parts covered by the given stripes.

Since power dissipation is an important issue for our architecture, these evaluations must be balanced by the power consumption associated to each algorithm. If the best results in terms of efficiency have been obtained with the RA+E algorithm and a 12x16 resolution, the corresponding power consumption is not the lowest of our tested configurations. Even if a 99.94% data amount reduction has been achieved, this algorithm remains more computationally expensive than the RA algorithm that requires fewer instructions. Since the latter gets very good results for the same resolution and the decimated macropixels configuration, it might thus be an interesting choice for our waking-up algorithm in terms of power consumption/efficiency compromise. The final choice depends on the hardware implementation and the application constraints.

Tests were performed to determine the response time, with 20x20 macropixels resolution. The average time between the arising of targets enter in the macropixels and the switching to high resolution, was respectively of 0.41s for the RA and 0.63s for the RA+E algorithms. As stated earlier, the RA+E algorithm indeed requires more computational steps. However, the larger macropixels sizes, the slower macropixels variations because of the fewer moving pixels contributions in blocks spatial averages. As a consequence, increasing macropixels size implies to lower the temporal resolution in order to get higher macropixels variations between each frame.

## 5. Chosen implementation

In order to perform both motion detection in power saving mode and tracking in active mode, an analogue programmable computational unit is considered. Analogue based computational system offer high compactness and low power consumption.

The considered computational unit is based on [13]. The SIMD machine presented (Figure 3) includes an NxM photosensors array to which an array of Nx(kM)

memory points is associated, where k is the number of memory elements per pixel. The so-formed matrix is bordered on one side by a vector of N processors. A column of multiplexers selects the column of pixels or memories to be used by the processor. A sequencer, implemented for example by a digital IP CPU, delivers the successive processors' instructions.

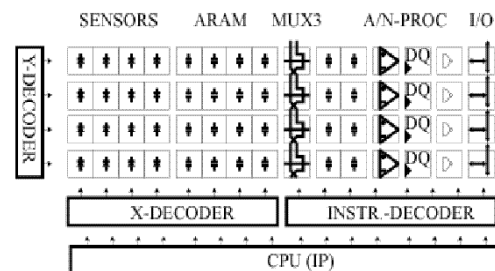


Figure 3 – Sensor architecture

The processor is a switched capacitor analogue computing unit shown on Figure 4. For each processor instruction, the switches configurations for the OTA and for the associated analogue registers are fixed. The multiplication (MAC) of an analogue value only requires 4 clock cycles with the following sequence of operations: writing on the left-most capacitor, clearing the other (weighted) capacitors and setting them into the feedback loop of the OTA, accordingly to the coefficient of the MAC.

The various operations required by the considered algorithms can be performed with this parallel architecture. Low-level real-time image processing for tracking in active mode is also possible. In addition, these processors can be set in idle mode when they are not being used.

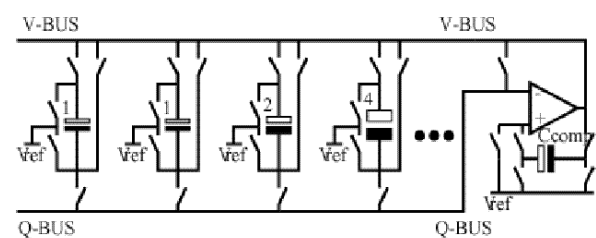


Figure 4 – Processing unit

The computational cost of the presented algorithms in power saving mode is proportional to the amount of data. This power saving must be balanced by the fact that performing a local spatial average has a cost in terms of power dissipation (charges and discharges of pixels capacitance), but which is quite negligible compared to power saving induced by the reduction of

data to compute (charges and discharges of bus capacitance and OTA).

For the Recursive Average algorithm, our architecture, working at 40MHz, requires 10 micro-instructions per pixel. Hence a computation time of 250ns per pixel. For the sequence used in our simulations, the mean power dissipation per micro-instruction is 14nW for a frame rate of 40fps.

With a 99.94% (12x16) data reduction from the analyzed full resolution scene (480x640), the total power dissipation would thus be of 27 $\mu$ W (192x140nW) for Recursive Average algorithm, as long as the system stays in power saving mode. For Recursive Average with Estimator algorithm, our architecture requires 20 micro-instructions per pixel. The corresponding power dissipation would thus be of 54 $\mu$ W.

## 6. Conclusion

Specific algorithms, which aimed at detecting motion with low spatial and temporal resolution, have been studied on a given video sequence. Original low resolution configurations like Decimated Macropixels or Uniform Stripes have been explored and a new algorithm giving excellent results has been presented. The considered image sensor will allow drastic power consumption reduction in the absence of motion, with a more than 99% reduction of the amount of processed data compared to full resolution images. A very interesting hardware implementation has also been proposed with very low power consumptions for the power saving mode

The proposed low resolution techniques should also be explored for image compression in order to remove spatial and temporal redundancy on static part of images.

Future works will incorporate tracking in high-resolution mode with actualization of ROI on targets (tracking) and switching from high resolution to low-resolution mode. For example, in order to reduce the processing unit load, local processing allowing inter-pixels cooperation for ROI actualization and in-pixel calculation of recursive average will be explored.

## 7. References

[1] Moini, A. and al., "An insect vision-based motion detection chip", *Solid-State Circuits*, IEEE, Volume 32, Issue 2, Feb. 1997 Page(s):279 – 284.

[2] Mehta, S.; Etienne-Cummings, R.; "Normal Optical Flow measurement on a CMOS APS imager", *Circuits and Systems*, ISCAS '04, IEEE, Volume 4, 23-26 May 2004 Page(s):IV - 848-51 Vol.4.

[3] Miliarud, V. and al., "A wide dynamic range CMOS active pixel sensor with frame difference", *Circuits and Systems*, ISCAS 2005, IEEE, 23-26 May 2005 Page(s):588 - 591 Vol. 1.

[4] Kemeny, S.E. and al., "Multiresolution image sensor", *Circuits and Systems for Video Technology*, IEEE, Volume 7, Issue 4, Aug. 1997 Page(s):575 – 583.

[5] Pain, B. and al., "Dynamically reconfigurable vision with high performance CMOS active pixel sensors (APS) Sensors", *Proceedings of IEEE*, Volume 1, 12-14 June 2002 Page(s):21 - 26 vol.1

[6] Saffih, F. and al., "Multiresolution CMOS Image Sensor", *Technical Digest of SPIE Opto-Canada 2002*, Ottawa, Ontario, Canada 9-10, May 2002, p.425.

[7] Massie, M. and al., "Vision chip for navigating and controlling Micro Unmanned Aerial Vehicles", *Circuits and Systems*. ISCAS '03. Volume 3, 25-28 May 2003 Page(s):III-786 - III-789 vol.3.

[8] McCarley, P.L. and al., "Large format variable spatial acuity superpixel imaging: visible and infrared systems applications", *SPIE 2004*, Orlando, FL.

[9] Coulombe, J. and al., "Variable resolution CMOS current mode active pixel sensor", *Circuits and Systems*, *Proceedings. ISCAS 2000 Geneva*, IEEE, Volume 2, 28-31 May 2000 Page(s):293 - 296 vol.2.

[10] Artyomov, E.; Yadid-Pecht, O.; "Adaptive multiple resolution CMOS active pixel sensor", *Circuits and Systems*, 2004. ISCAS '04, Volume 4, 23-26 May 2004 Page(s):IV - 836-9 Vol.4.

[11] Sung-Min Sohn and al., "A CMOS image sensor (CIS) architecture with low power motion detection for portable security camera applications", *Consumer Electronics*, IEEE, Volume 49, Issue 4, Nov. 2003 Page(s):1227 – 1233.

[12] J. Richefeu; A. Manzanera; "Motion Detection with smart sensor", 9th Congress Young Searchers in Computer Vision - ORASIS 2005, Clermont-Ferrand, May 24-27.

[13] Moutault, S.; Mathias, H.; Klein, J.-O.; Dupret, A.; "An improved analog computation cell for Paris II, a programmable vision chip", *Circuits and Systems*, 2004. ISCAS '04, Volume 5, 23-26 May 2004 Page(s):V-453 - V-456 Vol.5.