ADAPTIVE MULTIRESOLUTION FOR LOW POWER CMOS IMAGE SENSOR

Arnaud Verdant[1], *Antoine Dupret*[2], *Hervé Mathias*[2], *Patrick Villard*[1], *Lionel Lacassagne*[2]

[1] CEA LETI - MINATEC
17 rue des Martyrs, 38054 Grenoble Cedex 9 - France

[2] IEF
Bâtiment 220, Université de Paris 11, 91405 Orsay Cedex - France

ABSTRACT

To be implemented on an analog CMOS image sensor, a robust algorithm based on recursive operations is presented. It allows sensor's acuity adaptation to the scene activity. The main interest of the presented motion detection with adaptive thresholding is that, in a context of embedded steady camera, such a system allows focusing on targets with high resolution while keeping background in low resolution. Drastic power consumption reduction is achieved by tremendously reducing the amount of processed data.

Index terms— Motion detection, adaptive thresholding, tracking, multiresolution, CMOS image sensor.

1. INTRODUCTION

Visual tasks for embedded systems are confronted at the same time with high performance requirements and hard power consumption constraints. One way to address this issue is to design specific image processing architectures allowing some low level local analog processing to be performed at sensor's level (before A/D conversion), and thus be particularly power efficient. Thanks to submicron CMOS processes, the in-sensor processing can be performed without significantly impairing the device's resolution and sensitivity. However, specific adapted algorithms have to be developed concurrently. Since such sensors have to be fully autonomous, these algorithms have to be both robust and compliant to various environments while being at the same time computationally and power efficient.

In the case of embedded video surveillance, the physical implementation of motion detection is a particularly interesting task to investigate, since it allows extracting relevant information from a scene prior broadcasting. In addition, in some special applications where autonomy is an important concern, it could also be used to adapt sensor's performance and power consumption.

Tracking moving objects within the scene is also very interesting since it enables identifying the areas of interest in the image. If subsequent image processing is necessary on the detected objects, it will only be performed on the relevant parts of the image and not on the whole pixels. This will simultaneously save computation time and power because of a reduced set of pixels to be processed.

Among existing studies on motion detection and tracking, some particularly efficient methods have been proposed. Implementation of optical flow measurement is an interesting well-known technique, which has been explored in [1] and [2] for CMOS image sensors (CIS).

Image segmentation with difference to background and adaptive threshold has also been studied. In [3], motion detection is performed from recursive average computations and has been improved in [4] with a compensation of the trailing effect.

In [5], an efficient algorithm based on Σ-Δ modulation for artificial retinas is presented. For each pixel, background estimation and variance are computed with non-linear operations to perform adaptive local thresholding.

All the precedent approaches focus on optimizing motion detection but are not concerned with very low power image processing. Further power saving may be achieved by combining multiresolution with motion detection algorithm. Our approach, consisting in "waking up" the system when an event occurs in the scene, has also been explored in [6]. Low resolution is there achieved using decimated pixels. In [7], we have compared different low-resolution techniques for motion detection. The most efficient solution has been found to be computing spatial average of square areas.

In this paper we first briefly present our considered programmable analog architecture and the multi-resolution principle we proposed to save power. We then describe the proposed algorithms for motion detection, modified to include adaptive thresholding, and tracking. We then present results and compare them to a reference algorithm. Finally, an evaluation of the achieved power consumption is given.

2. SYSTEM ARCHITECTURE

2.1. Power consumption reduction and processing strategy

Systematic A/D conversion of all the matrix pixels, without taking into account relevant motion information, induces huge waste of power since many pixels are processed uselessly. Reducing both spatial and temporal resolution and thus the amount of data processed may also enhance power saving. Hence, combining specific motion detection algorithms with images of low resolution allows only A/D converting in high-resolution areas of interest, i.e. where some motion occurs, while static areas remain in low resolution. Spatial and temporal redundancy is so taken into account in A/D conversion. This leads to a particularly power-efficient video surveillance system.

Low power mode is achieved by reducing image resolution using macropixels. The macropixels are square blocks whose gray level is the spatial average of their constituting pixels. In low power mode, all pixels remain active but only to compute the macropixels value. This computation is performed at very low extra power consumption by sharing pixels sensor capacitance charges. Only the spatial average of macropixels is then taken into account to detect motion, with an improvement of results of 85% with respect to the decimated pixels solution [6].
In [7], we also analyzed the wake up function, i.e. switching from low resolution to high resolution when a sufficient variation of a macropixel gray level, indicating motion, is detected. If no motion is detected within a macropixel, its constitutive pixels values are neither converted nor sent out of the imager. Otherwise, the macropixel switches to high resolution together with a 5×5 macropixels neighborhood. All constitutive pixels of this ROI are then A/D converted and read out from the imager. Detecting macropixels variations is then equivalent to motion detection.
We are now considering the reverse transition when no motion is observed within a Region of Interest (ROI): a block of pixels (i.e. a macropixel) turns back to low power mode only if no motion is observed in the whole 5×5 neighborhood. This function implies to keep low resolution information in high resolution area in order to check averages variations. Computation is so performed on macropixels either in “standby” or “awake” mode (figure 1).

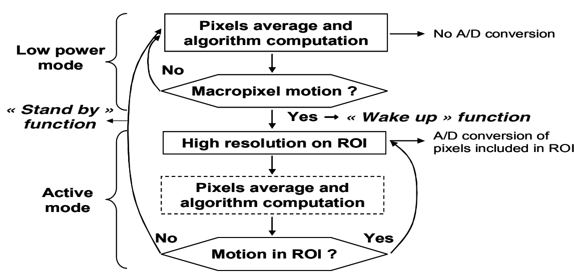

Figure 1. Sensor’s behavior (ROI: Region Of Interest)

Such a system allows both switching from active to low power mode but also having high resolution on a target before this one enters in a macropixel. A low power high-resolution tracking is so performed. Figure 2 shows the improvement obtained with our new method.

Figure 2. Improvement of target detection : a) image in full resolution; b) former results for multi resolution motion detection [7]; c) reduced target distortion obtained with new method

To face different situations found in video monitoring, an implementation on a versatile architecture is proposed.

2.2 Physical implemention and operators

In order to perform low power motion detection from pixels spatial average in a CIS, the implemented system should offer high compactness and low power consumption.
The considered programmable computational unit (figure 3) is a low power SIMD machine based on analog processing [8]. It is composed of an A×B photosensors array to which an array of A×(m B) analog memory points (Analog RAM) is associated, where m is the number of memory elements per pixel. Typically, m=3, A and B may be up to 1024. The so-formed matrix is bordered on one side by a vector of A switched capacitors analog processors. A column of multiplexers selects the column of pixels or memories to be used by the processor. A sequencer, implemented by a digital IP CPU, delivers the successive processors’ instructions.

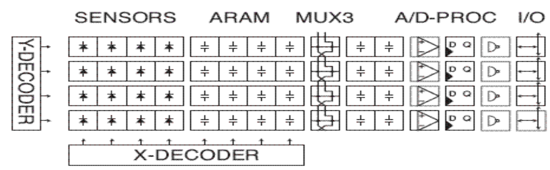

Figure 3. Sensor architecture

With such an architecture, pixel averaging can efficiently be performed by mixing capacitors charges at pixel level [9]. A digital implementation would require numerous computations and power consuming transfers of data.
The chosen architecture globally enables the implementation of “simple” algorithms at a much reduced power cost. “Simple” is to be understood as stepwise linear algorithms based on a reduced temporal or spatial kernel.

3. MOTION DETECTION ALGORITHM

3.1. Algorithm requirement

In our surveillance scheme, we aimed at performing motion detection with autonomous remote CIS sensors, in unknown environments. In such a configuration, algorithms must meet hard constraints of robustness and adaptability. Markovian algorithms are generally used to face these situations. However, they had to be simplified in order to satisfy the considered consumption and computational constraints while preserving their robustness.
As a reference algorithm, we consider the one presented in [5] which features non-intensive computation operations and high robustness. This Σ-Δ algorithm follows the Markov model used for real time implementations in [10]. The main improvements presented are a more robust detection with background estimation than with frame difference and local thresholding with no global computation.
Improvement of such an algorithm robustness is however required in our power saving strategy. Less false positives in motion detection induces keeping irrelevant static area in low resolution, implying thus lower power consumption.

3.2 Description of proposed algorithm

In order to be robust and adaptable to scene perturbations, our proposed algorithm is based on the computation of two recursive averages (*RA1* (1) and *RA2* (2)), each with its own time constant (fixed by the constants *N* and *M*): the slowest is used to bring out the background while the other, with short lag, filters the signal's fast perturbations. For each pixel, the main computation steps are described below:

$RA1_0 = S_0$ and $RA2_0 = S_0$

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

$$RA2_n = RA2_{n-1} - \frac{1}{M} RA2_{n-1} + \frac{1}{M} S_n \quad (2)$$

$$if \quad \Delta_n = \left| RA1_n - RA2_n \right| > k.\delta_n \rightarrow motion \quad (3)$$

where n represents the frame index, S_n the current gray level value for the considered block and $k.\delta_n$ a local threshold.

These recursive operations with few memory requirements make this algorithm easy to implement on our architecture. The time constant for fast recursive average (*RA1)* can be determined in order to allow an efficient fast perturbations filtering while not inducing significant trail effect. Considering the z-transform of the recursive average, the time constant is given by expression (4).

$$\frac{RA1(z)}{S(z)} = \frac{z}{N\left(z - \left(1 - \frac{1}{N}\right)\right)} = \frac{z}{N\left(z - e^{\frac{-T_e}{\tau}}\right)}, \ with \quad \tau = \frac{-Te}{\ln\left(1 - \frac{1}{N}\right)} \quad (4)$$

Motion is considered in (3) when Δ_n becomes larger than a local threshold $k.\delta_n$, which depends on Δ_n temporal activity. The adaptive threshold is obtained by amplifying δ_n that is the recursive average of Δ_n (5). With this method, $k.\delta_n$ directly depends on Δ_n perturbations levels, periodicity or persistence. Then, the corresponding and some neighboring blocks are switched to high resolution. Δ_n acts as a pass-band filter selecting only moving objects of interest in the scene.

$$\delta_n = \delta_{n-1} - \frac{1}{P}\delta_{n-1} + \frac{1}{P}\Delta_n \quad (5)$$

The time constant of this threshold must be quite slow in order to adapt sensitivity to persistent perturbation only.
In this algorithm, the four constants (*k, M, N, P*) depend on the to-be detected objects properties (mainly size and speed). However, knowing the type of object to detect, local adaptive threshold is assumed by our algorithm. For example, for the tested sequences (see paragraph 4.1), we chose: $N=2^2$; $M=2^4$; $P=2^6$; $k=1.8$.

In order to further reduce false positive detections induced by noisy elements of the scene, an activation function is used. Based on an a priori scenario, in addition to the algorithm steps, this function favors local translations. A macropixel is so allowed to switch to high resolution (activation) only if the local motion detection has been preceded by previous motion detection or activation in nearby macropixels in a given direction (figure 4).

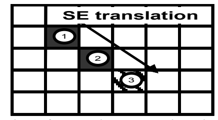

Figure 4. Local translation favouring. Motion is successively detected in macropixels 1 and 2. Only macropixel 3 can then switch to high resolution.

4. RESULTS

4.1 Algorithm performance

Simulations have been performed on MATLAB© for the considered algorithm with a macropixel-based low-resolution configuration. Different sequences representing a wide variety of indoor and outdoor conditions have been tested with different macropixels sizes: *Hall Monitor* (44×24 from 352×240, figure 2), *dtneu_schnee* (64×48 from 768×576, falling snow) and *kwbB* (i21www.ira.uka.de) *(*74×56 from 740×560), *Walk* (40×40 from 640×480, rustling foliage) (IEF's sequence), *Pets 2002* (64×24 from 640×240 flickering light) (respectively a, b, c, d in figure 5).

Figure 5. Sequences with full and low resolution

During the simulation, the state of each macropixel (high or low resolution) is compared to ground truth information for each frame. The number of true and false positives and negatives (TP, TN, FP, FN) can thus be counted.
We used the following motion detection performance metrics based on [11]: Detection Rate (DR=TP/(TP+FN)); False Alarm Rate (FAR=FP/(TP+FP)); and False Positive Rate (FPR=FP/(FP+TN)). A specific parameter (for power saving evaluation), Standby Rate (SB=(TN+FN)/number of macropixels) which gives the percentage of the image staying in low resolution, has been introduced. Table 1 and 2 show measured results for different gray level sequences, respectively for Σ-Δ algorithm [5] and for our new

algorithm. For each sequence, the data reduction percentage obtained with the chosen low resolution is indicated. Thus, for example, in the case of the 64×24 macropixels low resolution applied on the original 640×240 full resolution *Pets2002* sequence, (64×24) data are processed. A data amount reduction of 99% is so achieved, impacting a power reduction of similar ratio.

TABLE 1 - PERFORMANCE METRICS WITHOUT ACTIVATION FUNCTION (NF) AND WIITH ACTIVATION FUNCTION (F) WITH Σ-Δ ALGORITHM (N=3)

Grey level sequence (data reduction %)	Performance metrics (%)							
	DR		FAR		FPR		SB	
	NF	F	NF	F	NF	F	NF	F
Pets 2002 (99)	96.4	94.8	57.0	32.0	15.7	5.0	76.2	86.5
Hall (98.75)	94.3	92.0	8.3	5.8	1.4	0.8	86.6	88.3
kwbB (99)	99.1	98.3	10.3	8.2	1.0	0.7	91.6	92.4
dtneu_schnee (99.3)	99.8	99.7	70.2	60.8	44.0	28.5	47.7	60.9
Walk (99.5)	100	99.8	87.3	76.9	34.8	15.0	57.8	75.5

TABLE 2 - PERFORMANCE METRICS WITH OUR NEW ALGORITHM

Grey level sequence (data reduction %)	Performance metrics (%)							
	DR		FAR		FPR		SB	
	NF	F	NF	F	NF	F	NF	F
Pets 2002 (99)	94.6	92.7	15.9	16.5	2.1	1.9	88.4	89.7
Hall (98.75)	96.9	96.6	12.9	12.4	2.6	2.4	83.4	84.0
kwbB (99)	99.0	98.0	25.4	23.2	2.8	2.2	90.1	91.2
dtneu_schnee (99.3)	99.5	98.6	23.9	22.6	5.3	4.6	81.5	83.1
Walk (99.5)	98.7	94.5	64.4	20.1	6.8	0.7	83.3	89.4

These results show an equivalent DR for all sequences. Therefore, better results are obtained with our algorithm concerning FPR and FAR for the 3 sequences having the biggest perturbations *(Pets2002*, *Walk* and *dtneu_schnee)*. A similar efficiency is obtained for the *Hall* sequence whereas *kwbB* results are in favor of Σ-Δ algorithm. Actually, in *kwbB*, some camera oscillations come to create non relevant motion. Since more trailing effect is induced by our algorithm, more false positives are then generated.

The activation function, by finding local translations and filtering irrelevant motion, shows FAR improvement results. SB metric globally shows larger areas staying in standby mode in our algorithm, for the same DR than Σ-Δ algorithm. Since low resolution implies less power consumption, a better power saving can be expected with our algorithm than with Σ-Δ algorithm, without affecting detection efficiency.

4.2 Power consumption

In order to check our power saving concept, we compared the estimated power consumption induced by our motion detection scheme implemented on the presented analog architecture and its digital counterpart.

With a 32×24 macropixels resolution applied on a full 320×240 resolution scene and the analog architecture working at 40kHz, the total estimated power dissipation would be of 301µW (at 25fps) [12] as long as the system remains in power saving mode.

For the power performance evaluation of digital implementation, we chose a PowerPC G4 (7447) associated to a common image sensor acquiring full resolution images. The presented algorithm has been split into two parts. The first part, which concerns the macropixel generation from full resolution images, requires about 1 cycle per pixel to compute 8x8 pixels macroblocks. The second part, which concerns double recursive average filtering of macropixels values, requires 20 cycles per pixel with scalar operations and only two with SIMD Altivec instructions. Two images sizes have been considered: 256x256 and 348x240. For such sizes, the PowerPC is respectively ×610 and ×434 faster than real time (40ms for 25fps). That means we can “downclock” the processor frequency to nearly 2 MHz. In that case, the estimation of the power consumption of PowerPC is about 200mW to which the power consumption of about 10mW of the 320x240 sensor must be added.

5. CONCLUSION

We have presented an algorithm allowing low power motion detection on low resolution images using an already developed programmable architecture for CIS. An improvement of robustness compared to reference low level motion detection algorithm has been exposed, thus leading to lower power consumption considering our motion detection scheme. High resolution is applied on moving objects while keeping static parts in low resolution. A low power “pseudo-tracking” is so performed from only 1% of data with respect to full resolution. A power saving gain about 100 can so be expected. Future work will validate our approach, taking into account the impact of technological parameters on algorithm performances.

REFERENCES

[1] Moini A. and al., “An insect vision-based motion detection chip”, Solid-State Circuits, IEEE, Volume 32, Issue 2, Feb. 1997.

[2] Mehta S., Etienne-Cummings R., “Normal optical flow measurement on a CMOS APS imager”, Circuits and Systems, ISCAS '04, IEEE, Volume 4.

[3] Joo S., Zheng Q., “A temporal variance-based moving target detector”, IEEE Workshop PETS, Jan. 2005.

[4] Abdelkader M.F., Chellappa R., Qinfen Zheng, Chan A.L., “Integrated Motion Detection and Tracking for Visual Surveillance”, ICVS '06, IEEE.

[5] Manzanera A., Richefeu J., “A robust and computationally motion detection algorithm based on background estimation”, ICVGIP'04.

[6] Sohn S.M. 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.

[7] Verdant A., Dupret A., Mathias H., Villard P., Lacassagne L., “Low power motion detection with low spatial and temporal resolution for CIS”, CAMPS '06, IEEE.

[8] Moutault S., Mathias H., Klein J.-O., Dupret A., “An improved analog computation cell for Paris II, a programmable vision chip”, ISCAS '04.

[9] Saffih, F., Hornsey R., “Multiresolution CMOS image sensor”, Technical Digest of SPIE Opto-Canada 2002, Ottawa, May 2002, p.425.

[10] L. Lacassagne, M. Milgram, and P. Garda., “Motion detection, labeling, data association and tracking in real-time on risc computer”, *ICIAP*, 1999.

[11] Black J., Ellis T.J., and Rosin P., “A novel method for video tracking performance evaluation”, IEEE, VS-PETS, pages 125-132, October 2003.

[12] Verdant A., Dupret A., Mathias H., Villard P., “SystemC Validation of A Low Power Analog CMOS Image Sensor Architecture”, Newcas 2007.