DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-278

Trade-off between the number of bits per pixel and motion detection quality for a low power image sensor.

Camille Dupoiron, Arnaud Verdant, Gilles Sicard; DACLE/L3i, Université Grenoble Alpes, CEA-LETI, Minatec Campus, F38054, Grenoble, France

Abstract
Implementation of a motion detection algorithm in a very low power consuming image sensor is very constrained. A trade-off between the movement detection robustness and quantization level of pixel's signals for a grayscale image, had to be established. Simulations have been made for both quantization resolution and frame rate. Obtained results will help us design an optimized ultra-low power smart sensor.

Introduction
Real-time motion detection has become an inevitable step for image processing applications such as video surveillance, gesture recognition and more. For that purpose, several algorithms exist [1]-[5]. They are different in cost and in implementation. The more an algorithm is going to use complex operation, the more it will use power and occupy area on the chip. Some of them treat the information in its analog form [1]-[4] and others in their digital form [5].

For an image with 8 bits/pixel in grayscale only, the pixel value varies from 0 to 255. It seems important to highlight that the MSBs, most significant bits, hold the most important and structural elements of an image such as edges or brutal change of color, whereas LSBs hold the detail of the image such as texture, small variation of colors, weak edges and so forth. As it can be seen in Fig.1. on 8 bits most of the details are visible, when using only 2 bits to code the image signal, some details disappear and only the main edges are clear, such as the mirror edge.

Fig. 1: Quantization effect from 8bits/pixel (left) to 2bits/pixel (right)

As a consequence, in an ultra-low power consumption context, only the structural elements should be sufficient to get an accurate motion detection. Since the structural elements of the image are contained in the MSBs, one could focus on the minimum number of MSB necessary to get an accurate motion detection.

Objective
Objective of this work is to implement a motion detection algorithm in an ultra-low power image sensor. A trade-off between the movement detection quality and the number of bits per pixel, the quantization level, used to encode the signal for a grayscale raw image, had to be established. This study will help us design a computationally efficient low power smart sensor architecture.

Method
For a given dataset of a video sequence, well known motion detection algorithms have been tested, and their efficiency has been evaluated. For that purpose, two descriptors of efficiency are used [6]. The first one is the detection rate and second one is the false alarm rate. The detection rate measures how well the motion detection is performed. It represents the ratio of pixel correctly classified as a moving object over all the pixels which should have been classified as such. And the false rate is used to quantify how wrong the detection is, meaning that the pixel is considered as part of a moving object when it isn't. To calculate those two descriptors, the pixels can be classified in four different categories: the true positive, TP, the false positive FP, the true negative, TN, and the false negative FN.

$$d_r = \frac{N_{TP}}{N_{TP}+N_{FN}} \text{ and } f_a = \frac{N_{FP}}{N_{TP}+N_{FP}}$$

Where d_r is the detection rate, f_a the false alarm rate, and N_{TP}, N_{FN}, N_{FP} are respectively the number of true positives (TP), the number of false negatives (FN) and the number of false positives (FP). The true positives are in this case the pixel belonging to the moving object and classified as so by the algorithm. The false positive are, on the contrary, the pixels classified as belonging to the moving object when it is not actually the case. The true and false negatives are based on the same logic but for pixels which do not belong to the moving object. Those metrics are calculated for each frame and then an overall evaluation metric is found as their average over the entire video sequence.

This quality assessment is repeated until the image signal is coded on only 2bit/pixel. All simulations have also been made for different frame rates. The lower the frame rate would be, the more visible the movement would be and hence the detection rate should be higher. On the other hand the false alarm rate risks to also be higher.

This simulation has been performed for three different algorithms described below. Those algorithms have been choose for their simplicity to be implemented.

First algorithm: Frame difference
The first algorithm consists on a simple difference between two consecutive frames to get the motion. A moving object will not have the same position in frame i than in frame i+1, whereas the background, if the camera does not move, will stay the same. Some thresholding is also needed which would depends on the signal quantization level. However, the fixed threshold used in this algorithm limits the robustness.

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-278

Second algorithm: Manzanera Zipfian background estimation

This algorithm, based on sigma-delta modulation , is described by A. Manzanera in [7] and is called the Zipfian background estimation.

For each pixel of the frame, a background estimation M_n is performed from a delta modulation of the temporal signal. If the pixel value is N times higher than the difference between the intensity I_n of the pixel and the average M_n of the time series I_n at time n, the dispersion estimator V_n is decremented, otherwise V_n is incremented. In other words, the delta modulation of N times the abolute value of the difference between the signal and the estimated background is performed. Here N is taken equal to 3.The considered pixel belongs to a moving object if the dispersion estimator V_n is below the difference $\Delta_n = |M_n - I_n|$.

$for\ every\ frame\ n$:
$Compute\ the\ average\ of\ the\ time\ series\ I_n\ of\ the\ pixel$
$If\ I_n > M_{n-1}$
$\quad M_n = M_{n-1} + 1$
$If\ I_n < M_{n-1}$
$\quad M_n = M_{n-1} - 1$
$Compute\ the\ absolute\ difference$
$\Delta_n = |M_n - I_n|$
$Compute\ the\ \Sigma\Delta\ estimation\ of\ the\ absolute\ difference$
$If\ V_n > 3\Delta_n$
$\quad V_n = V_{n-1} - 1$
$If\ V_n < 3\Delta_n$
$\quad V_n = V_{n-1} + 1$
$Decide\ whereas\ the\ pixel\ belong\ to\ a\ moving\ object\ or\ not$
$If\ V_n < \Delta_n => motion\ detected$

This algoritm has some limitations, it will be inefficient for a moving object over a complicated background and for certain kind of motion such as remote objects with radial velocity [7].

Third algorithm: Motion feature extraction (MFE) [5].

The MFE algorithm generate self-speed adaptive motion features. It has been developped employing row-parallel and pixel-parallel architectures designed for digital pixel sensor. This algorithm is composed of two steps: the first one extracts static features from each frame and the second uses those features to extract motion features only when a certain amount of motion is detected.

To generate the merged significant edge maps (MSEM) for each frame, first, the local feature are extracted. Thanks to four 5 x 5-pixel filtering kernel convolved with a 5 x 5-pixel local image centered at each pixel site, edges for each direction are calculated. Then only the most prevalent directional edges are selected. For each direction, we then have a binary map of the corresponding pixels belonging to an edge. Second, the global features are extracted. To do so, only a predetermined percentage of significant edge flags out of all pixels that have larger gradient values than the rest , is selected. Eventually, the merged significant edge maps can be computed by taking logical OR of the four significant edge map. Fig.2 shows the MSEM generation.

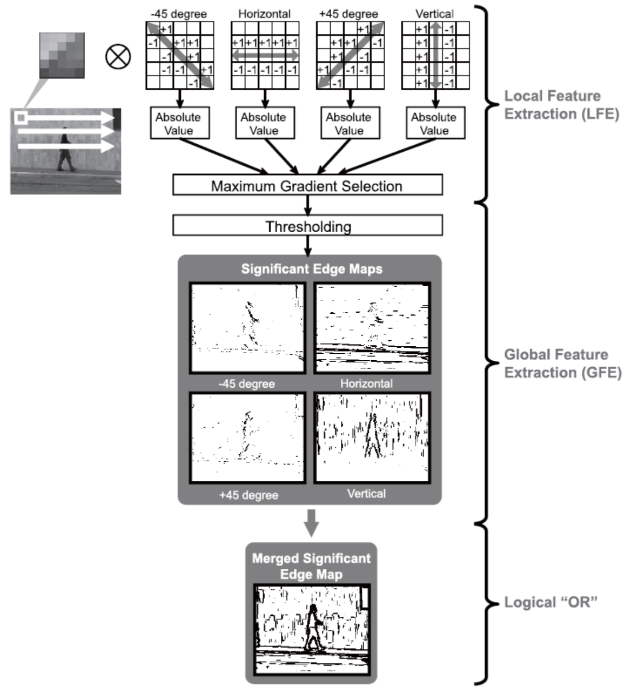

Fig.2: MSEM generation with LFE, GFE and logical "OR" operation. [5]

From the sequence of MSEMs, the motion features, MF, can be extracted. An initial accumulated edge map (AEM) is set as the first MSEM. A logical OR is applied between the AEM and the MSEM of the next frame. The count edge is performed and compared to a determined threshold: the motion detection threshold. If the count edge is lower than this threshold, a new AEM is obtained by taking logical OR between the MSEM from the next frame. The edge count is computed and compared to the threshold again. Otherwise, if the count edge is higher than this threshold, motion is detected and logical exclusive OR (XOR) is applied between the AEM and the last taken frame. The new AEM is set as the MSEM from the last taken frame. And those operation are performed until the end of the video sequence.

Hence motion features, are generated depending on the amount of motion occuring in the video sequence. Fig.3 shows the process of motion feature extraction using AEM and MSEM.

With those 3 algorithms, simulations to get the trade-off between, the number of bits/pixel and the efficiency of the motion detection can be done.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-278

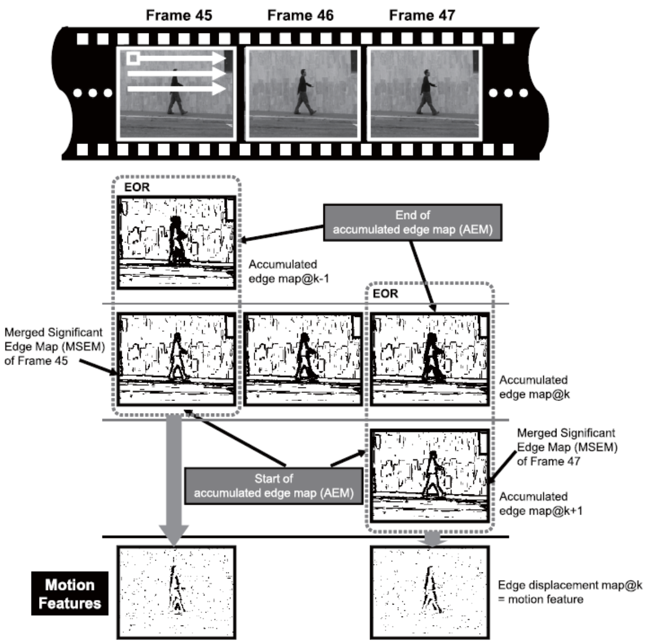

Fig.3: MFE process using AEM and MSEM [5]

Results

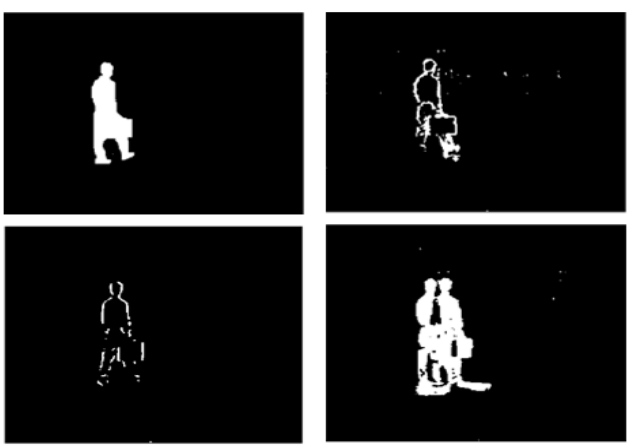

Figure 4: From left to right and up to down A– Ground truth of the moving object given in the dataset B- Movement detected for a quantization of 8bits with the first algorithm C- Movement detected for a quantization of 3bits with the first algorithm D- Movement detected for a quantization of 8bits and framerate divided by 9.

Using the image from the datasets shown in figure 7, the simulations have been made. For the sake of clarity, only the results from the Hall Monitor dataset are shown here, but the results are similar for both datasets. The results of figures 4, 5 and 6, show that in relation with the wanted image at the end of the detection algorithm (Fig.4.A, Fig.5.A and Fig.6.A. are the ground truth image), the detection using 8 bits/pixel seems better than with only 3 bits/pixel Fig.4.B & Fig.4.C (respectively Fig.5.B & Fig.5.C and Fig.6.B & Fig.6.C). In a nutshell, the lowest the quantization bit number gets, the worst the motion detection is. Fig.4.D. Fig.5.D and Fig.6.D show the effect of the frame rate diminution on the detection: the detection works but a lot of noise is induced.

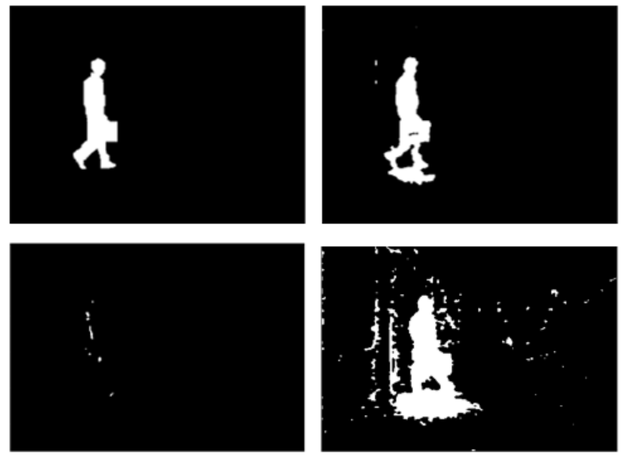

Figure 5: From left to right and up to down A– Ground truth of the moving object given in the dataset B- Movement detected for a quantization of 8bits with the second algorithm C- Movement detected for a quantization of 3bits with the second algorithm D- Movement detected for a quantization of 8bits and framerate divided by 9.

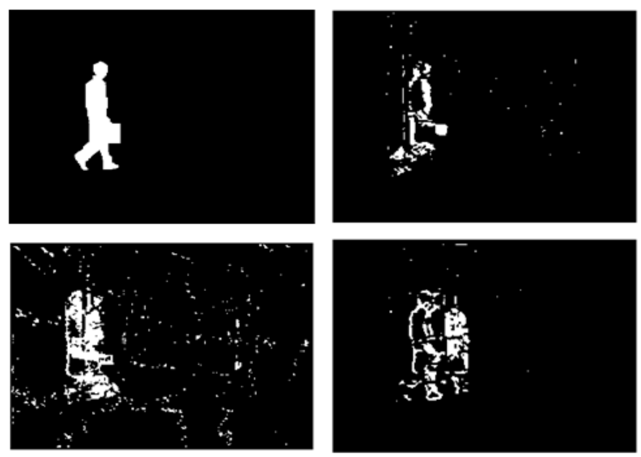

Figure 6: From left to right and up to down A– Ground truth of the moving object given in the dataset B- Movement detected for a quantization of 8bits with the third algorithm C- Movement detected for a quantization of 3bits with the third algorithm D- Movement detected for a quantization of 8bits and framerate divided by 9.

The results of figure 8 and 9, show that starting from 4 bits the detection rate is drastically falling. Between 5 and 8 bits of quantization, the detection rate curve has a weak slope and so the difference in performance is negligible. The false alarm rate follows the same pattern, the rate is more important between 1 and 4 bits of quantization but stays the same between 5 and 8 bits.
On figure 10, the results are not as obvious as on the last two figures since the detection rate increases significantly from 5 bits to 1 bit, so it would be tempting to says that if the pixel is coded on less than 5 bits/pixel the detection will be better. However the false alarm rate is also increasing from 5 bits to 1, hence the best detection acceptable with the less bits is around 5bits/pixel.

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-278

Two curves in Fig.8 and Fig.9, are slightly different than the other. They correspond to frame rates equal to 30 fps and 15 fps. They present a low detection rate but also a low false alarm rate. For the frame difference algorithm, the difference between the detection rate at 8bits/pixel for a frame rate of 30 fps (resp. 15 fps) is not so different from the detection rate at 5bits/pixel. However it is drastically different for the Zipfian background estimation algorithm (fig.8). Hence, for this sort of movement – a walking person – this algorithm will work better with a low frame rate.

To summarize, the results have been put in the tables 1 and 2. The detection and false alarm rate of the lower quantization level have been expressed relatively to the detection and false alarm rate of an 8 bits quantization level. In table 1, at 30 frames per seconds, algorithms 1 and 3 have an accurate detection for both 5 and 4 bits/pixels, however for algorithm 2, it would seems that at 30 frames per seconds, the most fitted level of quantization would be 6 bits/pixel. Yet, looking at table 2 one can notice that the most fitted quantization level for all algorithm is 5 bits/pixel. The detection is even better for the third algorithm at 5 bits/pixel than at 8 bits/pixel.

In conclusion, 5 quantization bits/pixel seems the more appropriate for this kind of detection.

Figure 7: left: Hall Monitor dataset image, right: PETS2006 dataset image.

fr = 30 fps	descriptors	8 bits	6 bits	5bits	4bits	2bits
Algorithm 1: Frame Difference	detection rate	1,00	0,94	0,86	0,74	0,03
	false alarm rate	1,00	0,98	0,97	0,95	0,85
Algorithm 2 : Zipfian Background Estimation	detection rate	1,00	0,67	0,38	0,13	0,00
	false alarm rate	1,00	0,27	0,16	0,23	0,42
Algorithm 3: Motion Feature Extraction	detection rate	1,00	1,08	1,07	1,16	1,00
	false alarm rate	1,00	0,92	1,04	1,00	1,59

Table 1: Frame rate of 30 fps, results of the simulations relatively to the case of 8 bits/pixel quantization level.

fr = 5 fps	descriptors	8 bits	6 bits	5bits	4bits	2bits
Algorithm 1: Frame Difference	detection rate	1,00	0,97	0,93	0,89	0,31
	false alarm rate	1,00	0,99	0,98	0,96	1,00
Algorithm 2 : Zipfian Background Estimation	detection rate	1,00	0,90	0,76	0,57	0,03
	false alarm rate	1,00	0,34	0,12	0,04	0,00
Algorithm 3: Motion Feature Extraction	detection rate	1,00	1,00	1,21	1,08	1,23
	false alarm rate	1,00	1,00	1,10	1,11	1,35

Table 2: Frame rate of 5 fps, results of the simulations relatively to the case of 8 bits/pixel quantization level.

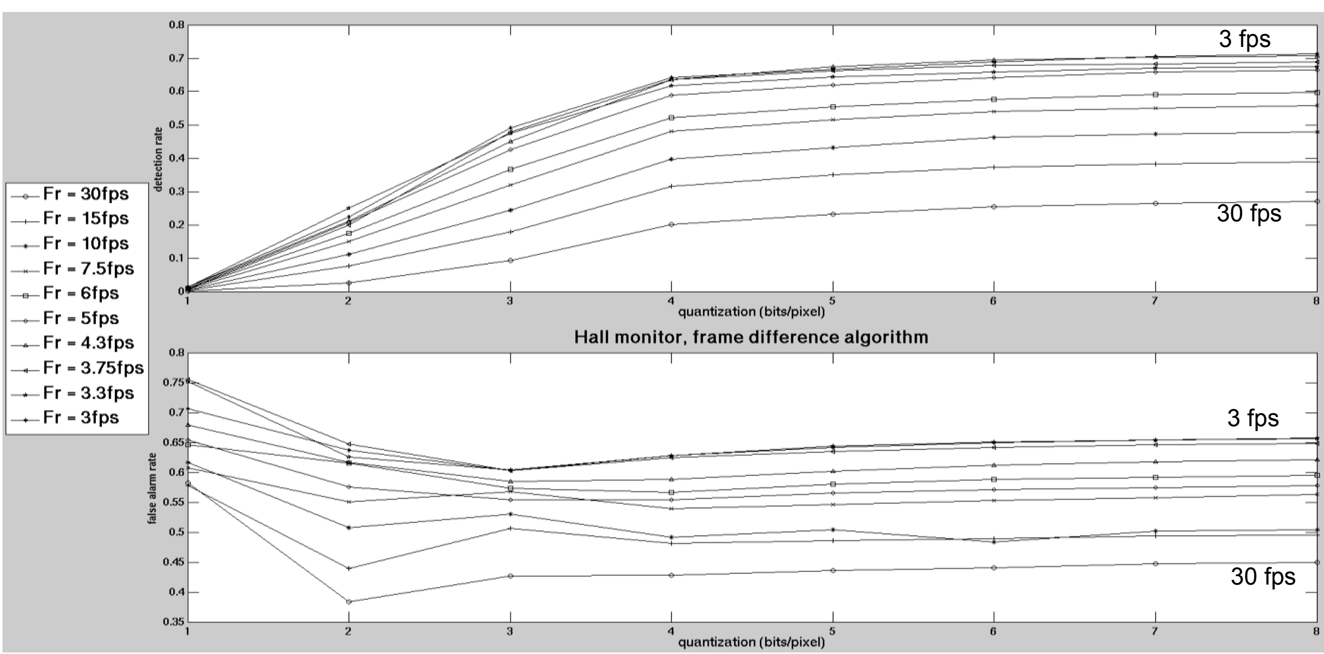

Figure 8: Up to down : for the first algorithm (frame difference), For each frame rate from 30 fps to 3 fps A- Detection rate depending on the quantization bit number. B- False alarm rate depending on the quantization bit number.

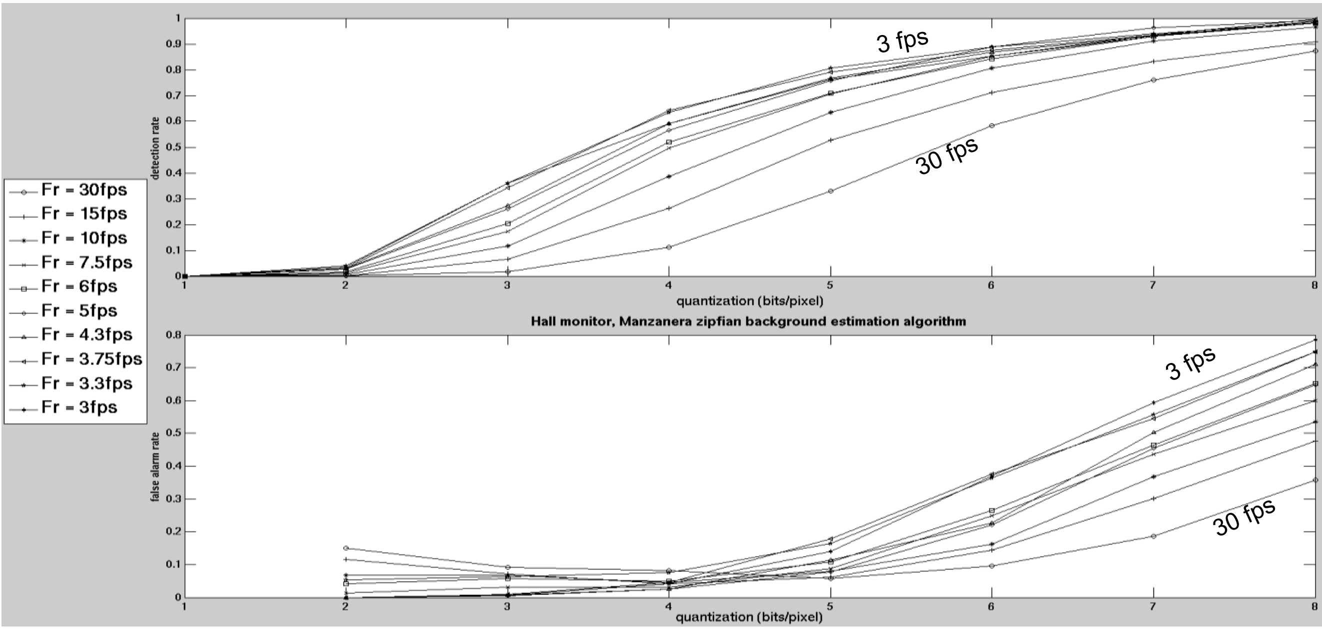

Figure 9: Up to down : for the second algorithm (Manzanera Zipfian Background Estimation). For each frame rate from 30 fps to 3 fps A- Detection rate depending on the quantization bit number. B- False alarm rate depending on the quantization bit number.

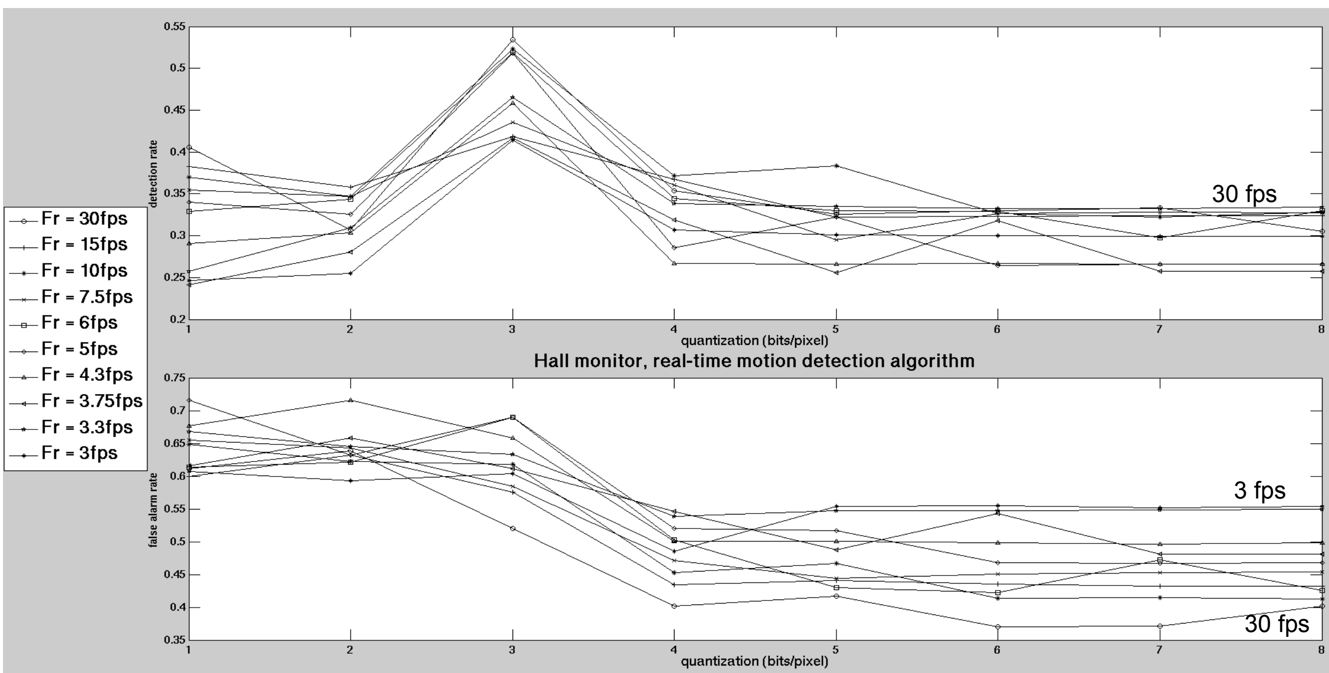

Figure 10: Up to down: for the third algorithm (Motion Feature Extraction). For each frame rate from 30 fps to 3 fps A- Detection rate depending on the quantization bit number. B- False alarm rate depending on the quantization bit number.

Limits

Several limits can be addressed to this study. Firstly, to test the algorithms, only two video sequence datasets have been tested, the Hall Monitor, presented here and a portion of the PETS2006 dataset whose results are the same as the Hall Monitor but they are not presented here. The Hall Monitor dataset is well known for motion detection since the movements are quite slow, there are few shadow effects and the video sensor is fixed.

Secondly, the choice of the algorithm will be indexed with the choice of the application for the image sensor. Hence, the given trade-off has to be validated in compliance with the application. And thirdly, the gamma correction has not been considerated in this study.

DOI: 10.2352/ISSN.2470-1173.2016.12.IMSE-278

Conclusion

To achieve low power, sensors of the SOA either reduce the frame rate or choose an adapted motion detection algorithm or have their data compressed at the end of the process. In our work, the ultra-low power aspect is the most important. To achieve this goal, in addition to use the optimized motion detection algorithm, and reduced frame rate, we also focus on using the least information possible; i.e. having the lowest number of bits to code the signal and still having efficient results in detection. This study with three different algorithms, has shown that, the optimized number of bits per pixel should be 5 bits/pixel for motion detection. Here, we place ourselve in an hardware implementation point-of-vue, only 5 bits/pixel will be send to the digital signal processing (DSP), this allows power consumption minimization. Nevertheless, the architecture will allow to test the configuration with a number of bits lower or equal to 5bits/pixel.

References

[1] Y. Muramatsu, S. Kurosowa, M. Furumiya, H. Ohkubo and Y. Nakashiba. "A signal-processing CMOS image sensor using a simple analog operation." *IEEE J. Solid-State Circuits* vol. 38 (Jan. 2003): pp. 101-106.

[2] V. Guev and R. Etienne-Cummings. "A pipelined temporal difference imager." *IEEE J. solid-State Circuits* vol.39 (Mar 2004): pp. 538-543.

[3] J. Choi, S. W. Han, S. J. Kim, S. I. Chang and E. Yoon. "A spatial-temporal multiresolution CMOS image sensor with adaptive frame rates for tracking the moving objects in region-of-interest and suppressing motion blur." *IEEE J. Solid-State Circuits* vol. 42 (Dec 2007): pp. 2978-2989.

[4] B. Zhao, X. Zhang, S. Chen, K-S Low and H. Zhuang. "A 64x64 CMOS image sensor with on-chip moving object detection and localization." *IEEE Trans. Circuits Syst Video Technol.* vol. 22 (Apr 2012): pp. 581-588

[5] H. Zhu and T. Shibata. «A real-time Motion-Feature-Extraction VLSI Employing Digital-Pixel-Sensor-Based Parallel Architecture.» IEEE Transctions on circuits and Systems for video Technology, vol. 24, no. 10, Oct. 2014

[6] N. Lazarevic-McManus, J. Renno, D. Makris and G.A. Jones. «Designing evaluation methodologies: the case of motion detection.» *In Ninth IEEE International Workshop on Performance Evaluation of Tracking and Surveillance.* 2006.

[7] Manzanera, Antoine. "Sigma-Delta Background Subtraction and the Zipf Law." *Progress in Pattern Recognition, Image Analysis and Applications (CIARP'07).* novembre 2007.

Author Biography

Camille Dupoiron received the Dipl.-Ing (M.Sc.) degree in electrical engineering from ENSEA School, Cergy, France in 2014 and the M.Sc. in electrical engineering from Illinois Institute of Technology (IIT), Chicago IL, USA, in 2014. She is currently working towards the Ph.D. degree in microelectronic and nanotechnology at CEA-LETI. Her current research interest includes ultra-low power CMOS smart sensors.