International Journal of Computer Science and Applications, 2008, Vol. 5, No. 3a, pp 1 - 19

A SMART ARCHITECTURE FOR LOW-LEVEL IMAGE COMPUTING

A. ELOUARDI, S. BOUAZIZ, A. DUPRET, L. LACASSAGNE, J.O. KLEIN, AND R. REYNAUD

Institut d'Electronique Fondamentale
Bat. 220. Université Paris-Sud XI
91405 Orsay, France

Abstract

This paper presents a comparison relating two different vision system architectures. The first one involves a smart sensor including analog processors allowing on-chip image processing. An external microprocessor is used to control the on-chip dataflow and integrated operators. The second system implements a logarithmic CMOS/APS sensor interfaced to the same microprocessor, in which all computations are carried out. We have designed two vision systems as proof-of-concept. The comparison is related to image processing time. Results reveal that one of the solutions to resolve the computational complexity of computer vision algorithms is to perform some low-level image processing on the sensor focal plane. This concept makes the vision systems more compact as a system on chip and increases performance thanks to the reduction of data-flow exchanges, with external circuits, and power consumption.

Keywords

Vision systems architecture; smart sensors; Image processing.

1. Introduction

The CCD technology have been the dominant tool for electronic image sensors during several decades due to their high photosensitivity, low fixed pattern noise (FPN), small pixel and large array sizes.

However, in the last decade, CMOS (Complementary Metal Oxide Semiconductor) sensors have gained attention from many researchers and industries due to their low energy dissipation, low cost, on chip processing capabilities and their integration on standard or quasi-standard VLSI (Very Large Scale Integration) process.

Still, raw output images acquired by CMOS sensors need further processing, mainly because of noise, blurriness and poor contrast. In order to tackle these problems, image-processing circuits are typically associated to image sensors as a part of the whole vision system. Usually, two areas coexist within the same chip for sensing and preprocessing that are implemented onto the same integrated circuit.

Robotics and intelligent vehicles need sensors with fast response time, low energy consumption, able to extract, from the environment, high-level information [1][2].

Adding hardware operators near pixels increases the computations potentiality and reduces inputs/outputs operations towards the central processor unit.

The integration of pixels array and image processing circuits on a single monolithic chip makes the system more compact and allows enhancing the behavior and the response of the sensor. Hence, to achieve low-level image processing tasks (early-vision), an artificial retina is a smart sensor which integrates analog and/or digital processing circuits in the pixel [3][4] or at the edge of the pixels array [5].

Moreover, this paper is built to get a conclusion on the aptitude of the retinas, as smart sensors, to become potential candidates for a system on chip, consequently to reach an algorithm-architecture adequacy.

We have done a comparison relating two different architectures dedicated for a vision system. The first one implements a logarithmic APS (Active Pixel Sensor) imager and a microprocessor. The second involves the same microprocessor with a CMOS artificial retina that implements hardware operators and analog microprocessors. We have designed two vision systems. The comparison is related to image processing time.

2. Image Processing Architectures: State of the Art

Different partitions for the architectural implementation of on-chip image processing with CMOS image sensors are proposed in [6]. The partition does not take into account only the circuit density, but includes also the nature of image processing algorithms and the choice of the operators integrated in its focal plane with the pixels. The difference between partitions is the location of the signal processing unit, known as a Processing Element (PE); this location becomes the discriminating factor of the different implementation structures:

The pixel processing, like the approach presented by P. Dudeck in [8], consists of one processing element (PE) per pixel. Each pixel typically consists of a photodetector, an active buffer and a signal processing element. The pixel-level processing promises many significant advantages, including low power as well as the ability to adapt image capture and processing to different environments during light integration. However, the popular use of this design idea has been blocked by the severe limitations on pixel size, the low fill factor and the restricted number of transistors in each PE.

In a view of great block partitioning, a global processing unit can be implemented beside the array of pixels. This way to do is one of the obvious integration methods due to its conceptual simplicity and the flexibility of the parameterization of the design features. Each PE is located at the serial output channel at the end of the chip. There are fewer restrictions on the implementation area of the PE, leading to a high fill factor of the pixel and a more flexible design. However, the bottleneck of the processing speed of the chip becomes the operational speed of the PE, and therefore, a fast PE is essentially required. The fast speed of the PE potentially results in the high complexity of the design [7] and the high power consumption of the chip [9].

Another structure is the frame memory processing. A memory array with the same number of pixels as the sensor is located below the imager array. Typically, the image memory is an analog frame memory that requires less complexity of design and processing time [10]. However, this structure consumes a large area, large power and high fabrication cost. Structures other than frame memory face the difficulty in implementing temporal storage. The frame memory is the most adequate structure that permits iterative operation and frame operation, critical for some image processing algorithms in a real time mode.

Even with these disadvantages, smart sensors are still attractive, mainly because of their effective cost, size and speed with various on-chip functionalities [11]. Simply, benefits exist when a camera with a computer are converted into a small sized vision system on chip (SoC).

3. Vision Systems Implementation

3.1. *An Artificial Retina Based Vision System (PARIS-ARM)*

3.1.1. *Sensor's architecture*

PARIS (Parallel Analog Retina-like Image Sensor) is an architecture model which implements, in the same circuit, an array of pixels, integrating memories, and an analog processors vector [12]. The architecture, shown in figure 1, allows a high degree of parallelism and a balanced compromise between communication and computations. Indeed, to reduce the area of the pixels and to increase the fill factor, the image processing is centered on a row of processors. Such approach presents the advantage to enable the design of complex processing units without decreasing the resolution. In return, because the parallelism is applied to a row of pixels, the computations which concern more than one pixel have to be processed in a sequential way. However, if a sequential execution increases the time of processing for a given operation, it allows more flexible process. With this typical readout mechanism of image, the column processing offers the advantages of parallel processing that permits high frequency and thus low power consumption. Furthermore, it becomes possible to chain basic functions in an arbitrary order, as in any digital SIMD (Single Instruction - Multiple Data) machine. The resulting low-level information extracted by the can be then processed by a microprocessor.

The array of pixels constitutes the core of the architecture. Pixels can be randomly read allowing windows of images or regions of interet (RoI). The selected mode, for the transduction of the light, is the integration mode. Two vertical bipolar transistors, associated in parallel, constitute the photosensor. For a given surface, compared to classic photodiodes, this disposal increases the sensitivity while preserving a large bandwidth [13] and a short response time can be obtained in a snapshot acquisition. The photosensor is then used as a current source that discharges a capacitor previously set to a voltage

Vref. In some cases, the semi-parallel processing imposes to store temporary results for every pixel in four MOS capacitors used as analog memories (figure 2). One of the four memories is used to store the analog voltage deriving from the photo-sensor. The pixel area is 50x50 µm² with a Fill Factor equal to 11%.

This approach eliminates the input/output bottleneck between extra-circuits even if there is a restriction on the implementation area, particularly for column width. Still, there is suppleness when designing the processing operators' area: the processing implementation is more flexible relatively to the length of the columns. Pixels of the same column exchange their data with the corresponding processing element (PE) through a Digital Analog Bus (DAB). So as to access any of its four memories, each pixel includes a bidirectional (4 to 1) multiplexer. A set of switches allows selecting the voltage stored in one of the four capacitors. This voltage is copied out on the DAB thanks to a bi-directional amplifier. The same amplifier is used to write the same voltage on a chosen capacitor.

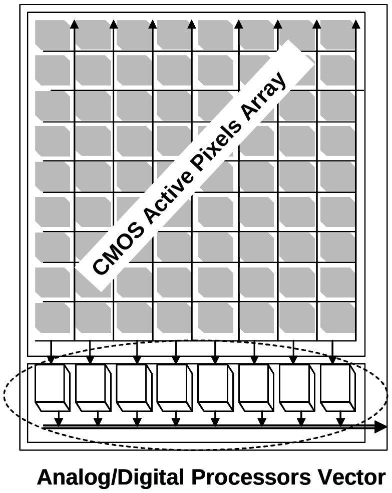


Fig. 1. Sensor's Architecture

The array of pixels is associated to a vector of processors operating in an analog/digital mixed mode (figure 3). In this paper, we shall detail only the analog processing unit: AP (figure 4). Each AP unit implements three capacitors, one OTA (Operational Transconductance Amplifier) and a set of switches that can be controlled by a sequencer.

Its functioning is much like a bit stream DAC: An input voltage set the initial charges in C_{in1}. The iterative activation of switches "mean" and/or "reset" reduces the amount of charges in C_{in1}. When "mean" is activated (C_{in1} and C_{in2} are connected together), and since C_{in1} and C_{in2} are at equal value, the charge in C_{in1} is divided by two. Iterating the operation N times, this step leads to a charge in C_{in1} of the form:

$$Q_{in1} = \langle C_{in1} \cdot V_{in1} \rangle / 2^N \tag{1}$$

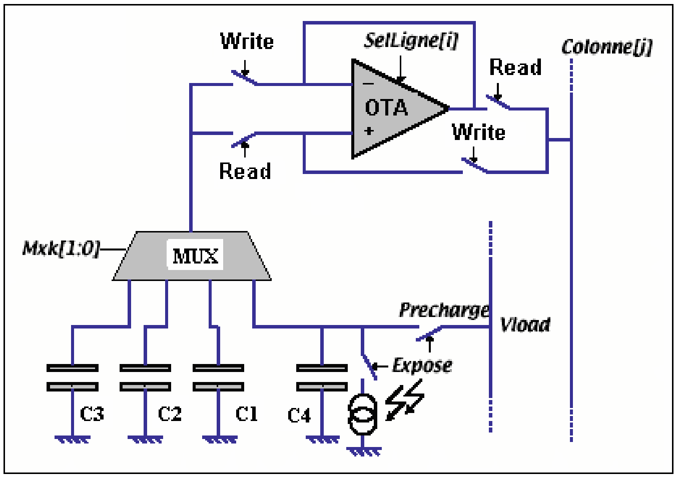
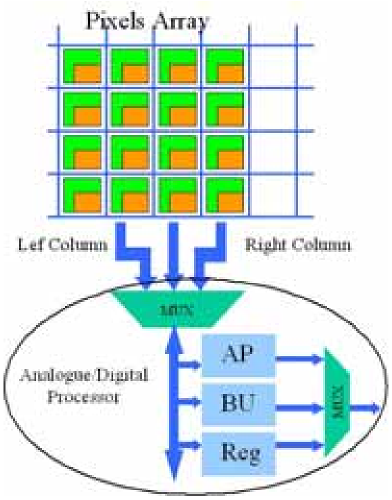


Fig. 2. Pixel Scheme (OTA: Operational Transconductance Amplifier)



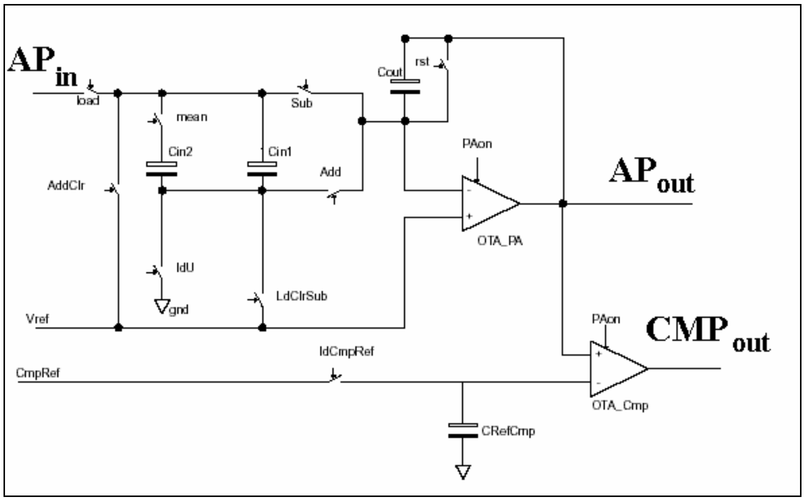
AP: Analog Processor
BU: Boolean Unit
Reg: Registers
Mux: Multiplexer

Fig. 3. Analog-Digital Processor (processing unit) Architecture

Thanks to the OTA, the remaining charge in capacitor C_{in1} is transferred to C_{out} when switch “Add”, or “Sub” are “On”. Therefore, the charges initially in C_{in1} are multiplied by a programmable fixed-point value. The capacitor C_{out} is so used as an accumulator that adds or subtracts charges flowing from C_{in1}. More detailed examples of operations can be found in [14].

In order to validate this architecture, a first prototype circuit has been designed including 16x16 pixels and 16 analog processing units. This first circuit allows validating the integrated operators through some image processing algorithms like edge and movement detection.

Using a standard 0.6 µm CMOS, DLP (Double-Layer Polysilicon) technology, this prototype “PARIS1” is designed to support up to 256x256 pixels. Considering this architecture and the technology used an artificial retina with higher resolution would lead to hard design constrain such on pixel access time and power consumption. To reduce costs the first prototype implements 16x16 pixels with 16 analog processors.



AP_{out}: Analog Processor Output **CMP_{out}:** Comparator output **AP_{in}**: Analog Processor input

Fig. 4. Analog Processor (AP) Unit Scheme

At a first order, the accuracy of the computations depends on the dispersion of the components values. The response dispersion between two AP units is 1%. The main characteristics of this chip are summarized in Table 1. Notice that the given pixel power consumption is its peak power; i.e. when the pixel is addressed. In other cases the OTA of the pixels are switched off and the pixel power consumptions is only due to C_4

resetting. In the same way, when the Processing Unit is inactive its OTA is switched off. Hence, the maximum power of the analog cells is: $C*[P_{pixel} + P_{Processing\ Unit}]$, where C is the number of columns, P_{pixel} and $P_{Processing\ Unit}$ are respectively the pixel power and is the processing unit power.

Table 1. Main characteristics of PARIS circuit

Circuit area (including pads)	10 mm²
Resolution (Pixels)	16x16
Number of APUs	16
Pixel Area	50x50 µm²
Area per Processing Unit	50x200 µm²
Clock Frequency	10 MHz
Processing Unit Power Consumption	300 µW
Row (16 Pixels) Power Consumption	100 µW

3.1.2. *Global Architecture (PARIS-ARM)*

We have designed a vision system (PARIS-ARM) based on PARIS1 retina, implementing converter DAC/ADC and a CPU core: the 16/32-bit ARM7TDMI[*] RISC processor. It is a low-power, general purpose microprocessor, operating at 50 MHz, developed for custom integrated circuits.

The embedded In-Circuit Emulator (ICE) is an additional hardware that is incorporated with the ARM core. Supported by the ARM software and the Test Access Port (TAP), it allows debugging, downloading, and testing software on the ARM microprocessor.

The retina, used as a standard peripheral of the microprocessor, is dedicated for image acquisition and low-level image processing.

With all principal components listed above, we obtain an architecture that uses a fully programmable smart retina. Thanks to the analog processing units, this artificial retina extracts the low-level information (e.g. edges detection). Hence, the system, supported by the processor, becomes more compact and can achieve processing suitable for real time applications.

The advantage of this architecture remains in the parallel execution of a consequent number of low level operations in the array integrating operators shared by groups of pixels. This allows saving expensive resources of computation, and decreasing the energy consumption. In term of computing power, this structure is more advantageous than that based on a CCD sensor associated to a microprocessor [15]. Figure 5 shows the global

[*] ARM System-on-Chip Architecture (2nd Edition), Steve Furber, September 2000.

architecture of the system and gives an overview of the experimental module (PARIS-ARM) implemented for test and measurements.

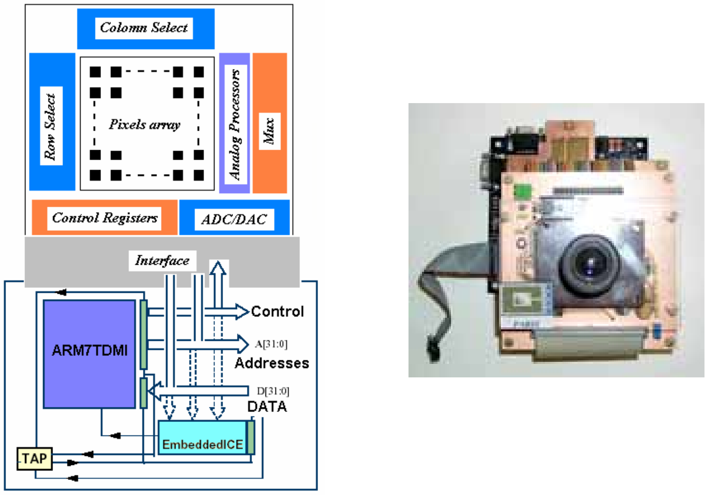


Fig. 5. Global architecture of PARIS1 based vision system and the Experimental module PARIS-ARM

3.2. *A Logarithmic CMOS Sensor Based Vision System (FUGA-ARM)*

In recent years CMOS image sensors have started to attract the attention in the field of electronic imaging that was previously dominated by charge-coupled devices (CCD). The reason is not only related to economic considerations but also to the potential of realizing devices with capabilities not achievable with CCDs. For applications where the scene light intensity varies over a wide range, dynamic range is a characteristic that makes CMOS image sensors attractive in comparison with CCDs [17]. An instance is a typical scene encountered in an outdoor environment where the light intensity varies over a wide range, as, for example, six decades. Image sensors with logarithmic response offer a solution in such situations. However, many works have been reported on high dynamic range of these logarithmic CMOS sensors having 130 dB like a dynamic [18][19].

Since the logarithmic sensors are non-integrating sensors (there is no control of the integration time), they can be an alternative to linear CMOS sensors. Due to their large dynamic range, they can deal with images having large contrast. This makes them very suitable for outdoor applications.

With to the random access, regions of interest (ROI) can be read out and processed. This reduces the image processing, resulting in faster image processing systems.

We have modeled a vision system (FUGA-ARM) whose architecture, shown in figure 6, is based on a logarithmic CMOS sensor (FUGA1000 figure 7), from FillFactory NV (Belgium) [20] and an ARM microprocessor (the same used for PARIS-ARM vision system).

The CMOS sensor (FUGA1000) is a 0.4528 inches (type-2/3") random addressable 1024x1024 pixels. It has a logarithmic light power to signal conversion. This monolithic digital chip integrates a 10 bit ADC and digital gain/offset control. It behaves like a 1 Mbyte ROM. After application of an X-Y address, corresponding to X-Y position of a pixel in the array, a 10 bit digital word corresponding to light intensity on the addressed pixel is returned.

Even if the sensor is really random addressed, pixels do not have a memory and there is no charge integration. Triggering and snapshot (synchronous shutter) are not possible.

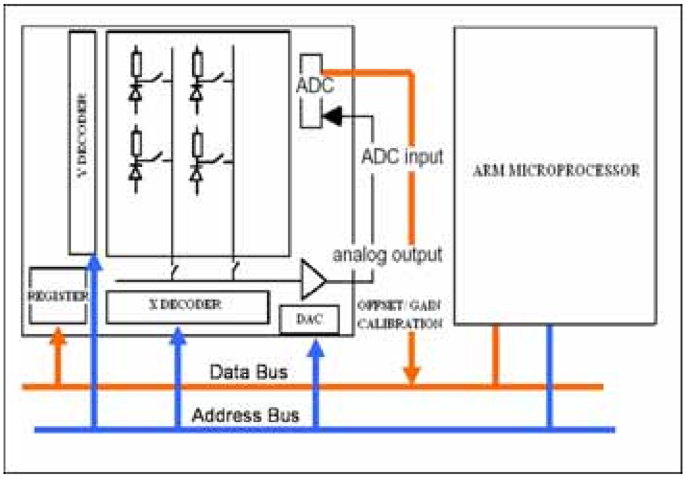


Fig. 6. Second architecture implementing a logarithmic CMOS sensor and an ARM7TDMI microprocessor.

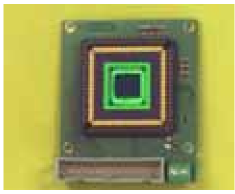


Fig. 7. The Logarithmic CMOS sensor (1024x1024 Pixels)

4. On-Chip and off-Chip Image Processing

4.1. *On Chip Image Processing*

The basis of the smart vision system on chip concept is that analog VLSI systems with low precision are sufficient for implementing many low-level vision algorithms for application-specific tasks. Conventionally, smart sensors are not general-purpose devices. They are specifically designed for dedicated applications.

Yet, in this paper, we do not wish to limit implementations to application-specific tasks, but also to allow this implementation to be used with general-purpose applications such as DSP[†] like image processors with programmability. The idea is based on the fact that many of the early level image processing operations, when used with general-purpose chips, are commonly shared with many image processors and do not require programmability. From the point of view of on-chip implementation, such algorithms are relatively pre-determined and fixed and their low precision can be compensated later by a back-end processing. Here, we will investigate what kind of image processing algorithms can be integrated on smart sensors as a part of early vision sequences and we will discuss their merits and the issues that designers should consider in advance.

General image processing consists of several image analysis processing steps: image acquisition, pre-processing, segmentation, representation or description, recognition and interpretation. This order can vary for different applications, and some stages of the processes can be omitted.

Local operation is also called mask operation where each pixel is modified according to the values of the pixel's neighbours (using kernel convolution). Denoting the pixels grey-levels at any location by P_{xy}, the response of a mask (3x3 kernel convolution as example) is given by equations (2) and (3):

$$O_{xy} = \sum_{i=1}^{3} \sum_{j=1}^{3} K_{ij} P_{x+i-2,y+j-2} \qquad (2)$$

$$K = \begin{bmatrix} \mathrm{k11} & \mathrm{k21} & \mathrm{k31} \\ \mathrm{k12} & \mathrm{k22} & \mathrm{k32} \\ \mathrm{k13} & \mathrm{k23} & \mathrm{k33} \end{bmatrix} \qquad (3)$$

[†] Digital Signal Processor

The grey-level P_{xy} of the pixel located at (x, y) position is replaced by O_{xy} value if the kernel mask is at the (x, y) location in the image. This computation is operated on each pixel moving the mask by one pixel location in the image at each step. Linear spatial filters are defined such that the final pixel value, O_{xy} , can be computed as a weighted sum of convolution mask (non-linear filters cannot be implemented in this way).

In the above case, a 3x3 local mask was taken as an example for the convolution mask. However, the size of the convolution mask can be expanded to 5x5, 7x7, 9x9, and larger, depending on the filter to be implemented.

For the on-chip integration with image sensors, local operations provide advantages of real time operation in acquisition and processing images, such as implementations of many practical linear spatial filters and image enhancement algorithms.

In order to understand the nature of a local operation and to find an adequate relationship between algorithms and on-chip implementations, we will look into the most usual algorithms, grouped according to the similarity of functional processing. The diagram presented in figure 8 allows understanding the functioning of such architecture (where each column is assigned to an analog processor). We choose a classical spatial filter example (a convolution with a 3x3 matrix). The Laplacian kernel L used is given by the matrix (4):

$$L = \begin{array}{|c|c|c|} \hline 0 & -1/4 & 0 \\ \hline -1/4 & 1 & -1/4 \\ \hline 0 & -1/4 & 0 \\ \hline \end{array} \qquad (4)$$

Pixels of the same row are simultaneously processed by the analog processor vector and the computations are iterated on image rows. The arithmetic operations are carried out in analog. The accumulation of the intermediate results is achieved in the analog processor using the internal analog registers. Starting from an acquired image, the figure 9 shows the L filtering operation result of an NxN pixels image, obtained by PARIS1 (N=16). Such operation is achieved in 6.8 ms. This computation time is globally due to: $\mathbf{T= N. (T_{add} + 4T_{div} + 4\ T_{sub})}$ where T_{add}, T_{div} and T_{sub} are the respective one pixel computation time, for an addition, a division and a subtraction operations. The computation time is proportional only to the number of rows and more elaborated algorithms can be implemented similarly.

Similar to averaging or smoothing, differentiation can be used to sharpen an image leaving only boundary lines and edges of the objects. This is a high pass filter. The most common methods of differentiation in image processing are the difference, the gradient and Laplacian operators. The difference filter is the simplest form of the differentiation subtracting adjacent pixels from the centred pixel in the horizontal and vertical directions.

The gradient filters represent the gradients of the neighbouring pixels (image differentiation) in forms of matrices. These gradient approaches and their mask implementations are represented with various methods: Robert, Prewitt, Sobel, Kirsch and Robinson methods [21].

The different local operations can be categorized into three major groups: smoothing filters, sharpening filters and Laplacian edge detection filters. Examples of the local operation algorithms are described in [22]. We have successfully implemented and tested a number of algorithms, including convolution, linear filtering, edge detection, motion detection and estimation. Some examples are presented below. Images are processed with different values of luminosity using the exposure time self calibration. Figure 10 gives examples of processed images in a luminosity range of (10-1000 Lux).

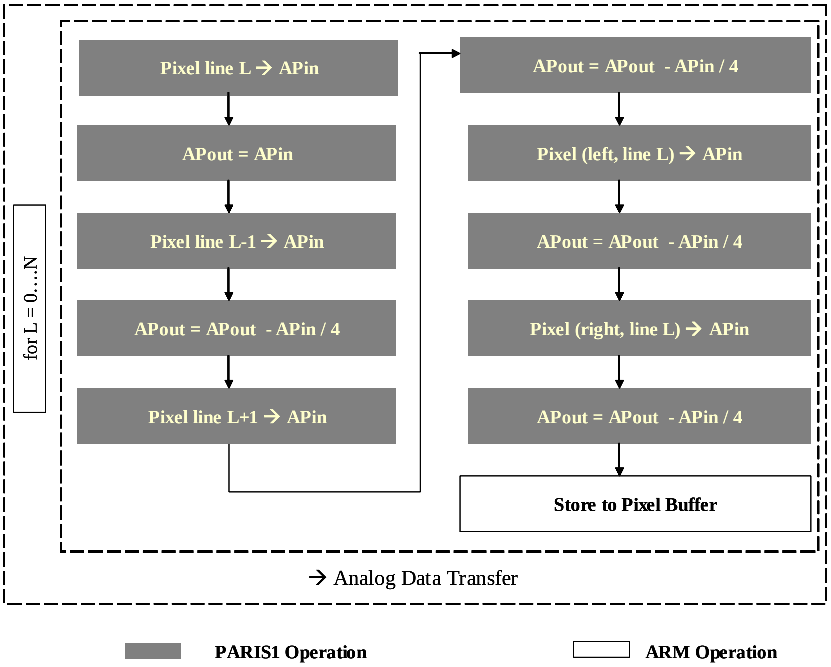


Fig.8. Diagram of the L filter operation

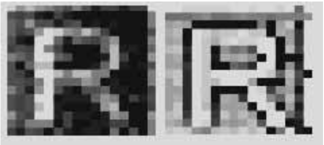
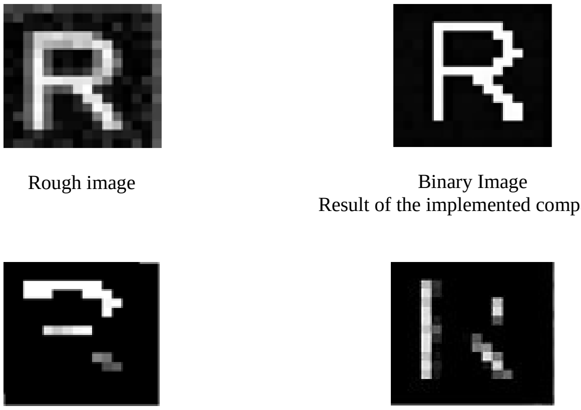


Fig.9. Original image (left) and filtered image (right)



Rough image

Binary Image
Result of the implemented comparator

Vertical Sobel operation
Result of the analog processor vector

Horizontal Sobel operation
Result of the analog processor vector

Fig.10. Examples of processed images

4.2. *Off Chip FPN Correction and Image Processing:*

The major drawback of the logarithmic sensor is the presence of a time-invariant noise in the images. The Fixed Pattern Noise (FPN) is caused by the non-uniformity of the transistors characteristics. In particular, threshold voltage variations introduce a voltage-offset characteristic for each pixel. The continuous-time readout of a logarithmic pixel makes the use of Correlated Double Sampling for the suppression of static pixel-to-pixel offsets quite impossible. As a result, the raw image output of such a sensor contains a large overall non-uniformity.

The downstream system of the sensor is then used to compensate the FPN: as the FPN is static in time, a simple look-up table with the size of the sensor's resolution can be used for a first-order correction of each individual pixel. Higher-order corrections can be employed when the application demands higher image quality. The FPN noise is removed from the images by subtracting to each pixel value the corresponding offset.

For the CMOS/APS sensor, the FPN correction is performed by the ARM microprocessor in real time and it is transparent (this operation can be achieved by an FPGA circuit for example). The sensor is shipped with one default correction frame. Figure 11 shows an image with the FPN and gives image after FPN correction.

The response of the logarithmic CMOS sensor typically is expressed as 50 mV output per decade of light intensity. After first order FPN calibration and using an ADC, a response non-uniformity of below 2mV remains, being quite constant over the optical range. This non-uniformity translates to about 4% of a decade. The temporal noise of the logarithmic sensor is 0.2 mV RMS.

(a) (b)

Fig. 11. (a) Image with FPN. (b) Image with removed FPN

For the FUGA-ARM vision system, images are processed on the ARM microprocessor. We established several algorithms of image processing similar to those established for PARIS-ARM vision system. Other more complicated algorithms which require diversified computing with exponential power were also established with the FUGA/ARM system. We recall that to carry out comparisons relating to the processing times, we chose to use the same processor (ARM7TDMI) for the different implemented systems.

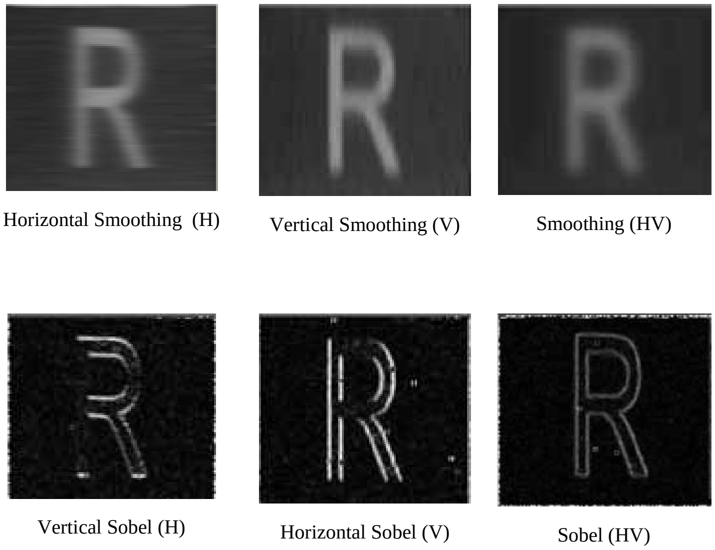
The filter we used has been designed by Federico Garcia Lorca [23]. This filter is a simplification of the Deriche filter [24], the recursive implementation of the optimal Canny filter. The smoother is applied horizontally and vertically on the image, in a serial way. Then a derivator is applied. Garcia Lorca derivator is, after simplification of Deriche, derivator, a 3x3 convolution kernel instead of a recursive derivator.

$$y(n) = (1-\lambda)^2.x(n) + 2\lambda.y(n-1) - \lambda^2.y(n-2) \tag{5}$$

$$y(n) = (1-\gamma)^2 x(n) + 2\gamma y(n-1) - \gamma^2 y(n-2) \tag{6}$$

with $\gamma = e^{-\alpha}$ (7)

$x(n)$ is the pixel source value. $y(n)$ is the pixel destination value and n is the pixel index in a one dimensional table representing the image. γ is an exponential parameter allowing much more filtering flexibility, depending on the noise within the image . If the image is very noisy we use a very smoothing filter: $\alpha = [0.5,0.7]$ otherwise we use bigger values of α : $\alpha = [0.8,\ 1.0]$. Figure 12 gives examples of smoothing and derivator filters implemented with the FUGA-ARM vision system and applied to 120x120 pixels images.


Horizontal Smoothing (H)

Vertical Smoothing (V)

Smoothing (HV)

Vertical Sobel (H)

Horizontal Sobel (V)

Sobel (HV)

Fig. 12. Examples of Image Processing Implemented with the FUGA1000 Sensor Based Vision System

5. Time Comparison in Image Processing

The aim is to compare the vision system implementing the logarithmic CMOS imager (FUGA-ARM) with the one based on PARIS retina (PARIS-ARM). This comparison is related to image processing time and don't take into account the exposure time for which we developed a continuous auto-calibration algorithm that can manage this state for our vision system. This avoids pixels saturation and gives an adaptive amplification of pixels output, which is necessary to the post-processing.

The calibration concept is based on the fact that when using a photo-sensor in an integration mode, a constant luminosity leads to a voltage drop that varies according to the exposure time. If the luminosity is high, the exposure time must be decreased, on the other hand if the luminosity is low the exposure time should be increased. One should then aim at lower exposure time to have faster image processing algorithms.

We have used a Laplacian edge detection algorithm and a Sobel filter algorithm to take several measurements of the computation times relating to the two architectures described bellow. For the artificial retina based system, these computations are carried out by the analog processors integrated on-chip. For the FUGA-ARM system, these computations are carried out by the ARM microprocessor.

The two computation time graphics presented in the figure 13 translate the diverse computing times for different square sensor pixel resolutions for both systems. It is significant to note that the acquisition time of the frames is not included in these measurements in order to evaluate just the data processing computing time. Times relating to the PARIS artificial retina were obtained by extension of the data processing timing obtained from those of the first prototype [12].

We deduce that the computation time for the FUGA-ARM system varies according to the pixels number N² (quadratic form). Hence, the computation time for PARIS-ARM system varies according to the rows number N (linear form) thanks to the analog processor vector. The equation (8) gives the definition of the CPP (Cycle Per Pixel) of a processor.

F_{CLK} is the processor frequency, T is the time computing, L is the rows number and C is the columns number:

$$CPP = \frac{T * F_{CLK}}{L * C} \qquad (8)$$

Figure 14 shows the evolution of the CPP for PARIS-ARM system and FUGA-ARM system.

Consequently, the microcontroller of the FUGA-ARM system carries out a uniform CPP relative to regular image processing independently of the number of proceeded pixels. For PARIS-ARM system, the CPP factor is inversely proportional to the rows number (N).

As a result, our implementation demonstrates the advantages of the single chip solution. Applications involving image processing algorithms will be less complex and efficient especially for high resolution.

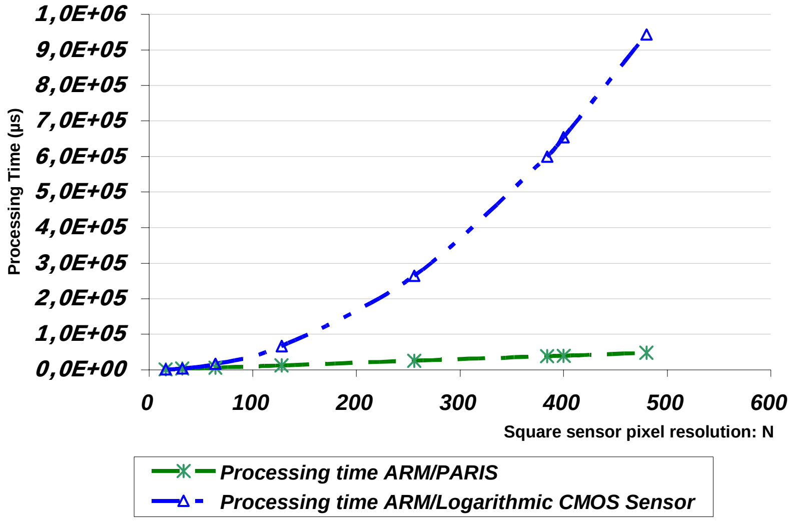


Fig. 13. Processing time of an edge detection: PARIS-ARM architecture versus ARM/Logarithmic CMOS sensor

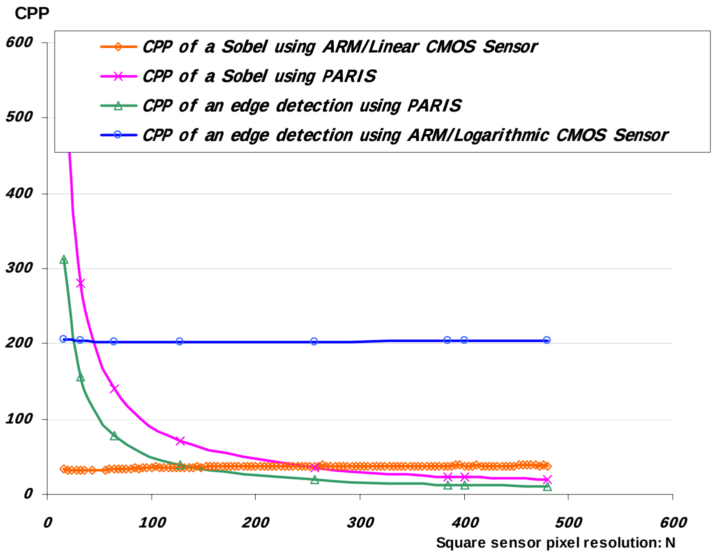


Fig. 14. Evolution of the CPP (Cycle Per Pixel) for PARIS-ARM and the FUGA-ARM architectures

6. Conclusion

When we wish to carry out real time image acquisition and processing, the hardware processing implementation with smart sensors becomes a great advantage. This paper presents one experience of this concept named a retina.

It is concluded that on-chip image processing with retinas will offer benefits of fast and parallel processing. Since each vision algorithm has its own applications and design specifications, it is difficult to predetermine optimal design architecture for every vision algorithm. However, in general, the column structures appear to be a good choice for typical image processing algorithms.

We have presented the architecture and the implementation of a smart integrated artificial retina based vision system. The goal is the integration of a microprocessor in the artificial retina to optimise the implemented hardware operators. Hence, designers and researchers can have a better understanding of smart sensing for intelligent vehicles [25].

We propose implementing such a system with high resolution in a complex application: intelligent vehicle embedding smart sensors for autonomous collision avoidance and objects tracking.

References

[1] Juergen Kogler, Hannes Hemetsberger, Bram Alefs, Wilfried Kubinger and William Travis, “Embedded Stereo Vision System for Intelligent Autonomous Vehicles”. IEEE Intelligent Vehicles Symposium 2006, pp 64-69. Tokyo, Japan, June 13-15, 2006.

[2] Juan M. Collado, Cristina Hilario, Arturo Escalera and Jose M. Armingol, “Self-calibration of an On-Board Stereo-Vision System for Driver Assistance Systems”. Proceedings of IEEE Intelligent Vehicles Symposium 2006, pp 156-162. Tokyo, Japan, June 13-15, 2006.

[3] D. Stoppa, A. Simoni, L. Gonzo, M. Gottardi, and G. Betta, "A 138 dB dynamic range CMOS image sensor with new pixel architecture." IEEE International Solid State Circuits Conference, pp. 40-41, 2002.

[4] Abbas El Gamal and al, "Pixel Level Processing: ¬Why, what and how?". SPIE, Vol.3650, pp. 2-13, 1999.

[5] Y. Ni, J.H Guan, "A 256x256-pixel Smart CMOS Image Sensor for Line based Stereo Vision Applications", IEEE, J. of Solid State Circuits, Vol. 35 No. 7, pp. 1055-1061, Juillet 2000.

[6] Alireza Moini, "Vision chips or seeing silicon", Technical Report, Centre for High Performance Integrated Technologies and Systems, The University of Adelaide, March 1997. Kluwer Academic Publishers, ed. I. 0-7923-8664-7. 2000.

[7] S. Chen, A. Bermak, W. Wan and D. Martinez, “A CMOS image sensor with combined adaptive-quantization and QTD-based on-chip compression processor”. IEEE Custom Integrated Circuits Conference. San Jose, California, USA. September 2006.

[8] P. Dudek, "A programmable focal-plane analog processor array”, Ph.D. thesis, University of Manchester Institute of Science and Technology (UMIST), May 2000.

[9] Miguel Arias-Estrada. "A Real-time FPGA Architecture for Computer Vision", Journal of Electronic Imaging (SPIE - IS&T), Vol. 10, No. 1, pp. 289-296, January 2001.

[10] Z. Zhou, B. Pain, E. Fossum, “Frame-transfer CMOS Active Pixel Sensor with pixel binning,” IEEE Trans. On Electron Devices, Vol. ED-44, pp.1764-8, 1997.

[11] D. Seguine, “Just add sensor - integrating analog and digital signal conditioning in a programmable system on chip,” Proceedings of IEEE Sensors, vol. 1, pp. 665–668, 2002.

[12] A. Dupret, J.O. Klein, A. Nshare, "A DSP-like Analog Processing Unit for Smart Image Sensors", International Journal of Circuit Theory and Applications. V30: p. 595-609. 2002.

[13] A. Dupret, E. Belhaire, and J.-C. Rodier, "A high current large bandwidth photosensor on standard CMOS Process". EuroOpto'96, AFPAEC, Berlin, 1996.

[14] A. Dupret, J.O Klein, A. Nshare “A programmable vision chip for CNN based algorithms. IEEE International Workshop on Cellular Neural Networks and Their Applications. CNNA, Catania, Italy. May 2000.

[15] Dave Litwiller, “CCD vs. CMOS: Facts and Fiction”. PHOTONICS SPECTRA, Laurin Publishing Co. Inc. January 2001.

[16] Dave Litwiller, “CCD vs. CMOS: Maturing Technologies, Maturing Markets”. PHOTONICS SPECTRA, Laurin Publishing Co. Inc. August 2005.

[17] B. Dierickx, J. Bogaerts, "Advanced developments in CMOS imaging", Fraunhofer IMS workshop, Duisburg, 25 May 2004.

[18] M. Loose, K. Meier, J. Schemmel, “CMOS image sensor with logarithmic response and self calibrating fixed pattern noise correction”. Proc. SPIE Vol 3410, pp.117-127. ISBN 0-8194-2862-0. 1998.

[19] B.Choubey, et. al.,”An Electronic-Calibration Scheme for Logarithmic CMOS Pixels,” IEEE Sensors Journal, Vol.6, No. 4,August 2006.

[20] FUGA1000-EVB from FillFactory: http://www.framos.co.uk/news/newsletter/Notes0301.pdf

[21] Handbook of Image & Video Processing. Edited by Al Bovik. Academic Press, San Diego, California, 2000.

[22] Gerard Blanchet, Maurice Charbit, “Digital Signal and Image Processing Using Matlab”. Iste Publishing Company. Javier 2006.

[23] R. Deriche, "Fast algorithms for low level-vision". IEEE Transaction of Pattern Analysis and Machine Intelligence, vol 12-1, 1990.

[24] F. Garcia Lorca, L. Kessal, D. Demigny "Efficient ASIC and FPGA implementation of IIR filters for real time edge detections", Proc. International Conference on Image Processing, IEEE ICIP 1997.

[25] A. Elouardi, S. Bouaziz, A. Dupret, L. Lacassagne, J.O. Klein, R. Reynaud, “A Smart Sensor for Image Processing: Towards a System on Chip”. IEEE International Symposium on Industrial Electronics ISIE’06, p-p 2857-2862. Montréal, Canada. July 9-12, 2006.