IOP Publishing Measurement Science and Technology

Meas. Sci. Technol. **18** (2007) 2817–2826 doi:10.1088/0957-0233/18/9/011

Time comparison in image processing: APS sensors versus an artificial retina based vision system

A Elouardi, S Bouaziz, A Dupret, L Lacassagne, J O Klein and R Reynaud

Fundamental Electronics Institute, Bat. 220, Paris XI University, 91405 Orsay, France

E-mail: elouardi@ief.u-psud.fr

Received 23 November 2006, in final form 6 June 2007
Published 20 July 2007
Online at stacks.iop.org/MST/18/2817

Abstract

To resolve the computational complexity of computer vision algorithms, one of the solutions is to perform some low-level image processing on the sensor focal plane. It becomes a smart sensor device called a retina. This concept makes vision systems more compact. It increases performance thanks to the reduction of the data flow exchanges with external circuits. This paper presents a comparison between two different vision system architectures. The first one involves a smart sensor including analogue 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.

Keywords: instrumentation, vision system architecture, retinas, CMOS/APS sensors

(Some figures in this article are in colour only in the electronic version)

1. Introduction

Intelligent vehicles and robots need sensors with fast response time, low energy consumption and the ability to extract high-level information from the environment [1, 2]. Adding hardware operators near pixels increases the computations potentiality and reduces inputs/outputs operations towards the central processor unit.

CCD technology has been the dominant tool for electronic image sensors for 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 drew attention from many researchers and industries due to their low energy dissipation, low cost, on chip processing capabilities and their integration on a 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 with 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.

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

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

We have made a comparison between two different architectures dedicated to 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 analogue microprocessors. We have designed two vision systems. The comparison is related to image processing time.

2. Review of integrated operators on smart sensors

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 also includes 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.

Pixel processing, like the approach presented by 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. 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 severe limitations on pixel size, the low fill factor and the restricted number of transistors in each PE.

In view of great block partitioning, a global processing unit can be implemented beside the array of pixels. This is one of the obvious integration methods due to its conceptual simplicity and the flexibility of the parametrization 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 high complexity of the design [7] and high power consumption of the chip [9].

Another structure is 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 analogue 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 the frame memory face 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 is converted into a small sized vision system on a chip (SoC).

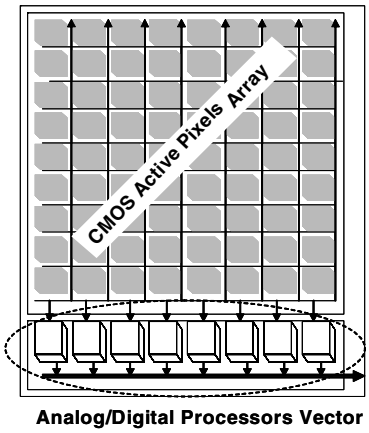


Figure 1. Sensor architecture.

3. Systems description

3.1. An artificial retina based vision system (PARIS-ARM)

3.1.1. Sensor architecture. A PARIS (parallel analogue retina-like image sensor) is an architecture model which implements, in the same circuit, an array of pixels, integrating memories and an analogue processor 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 centred on a row of processors. Such an approach presents the advantage of enabling 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 a more flexible process. With this typical readout mechanism of an image, column processing offers the advantages of parallel processing that permit 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 this can then be 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 interest (RoI). The selected mode, for the transduction of 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 V_{ref}. Semi-parallel processing can use the four MOS capacitors, integrated in each pixel, as analogue memories to store temporary results (figure 2). One of the four memories is used to store the analogue voltage derived from the photo-sensor. The pixel area is $50 \times 50\ \mu\mathrm{m}^2$ with a fill factor equal to 11%.

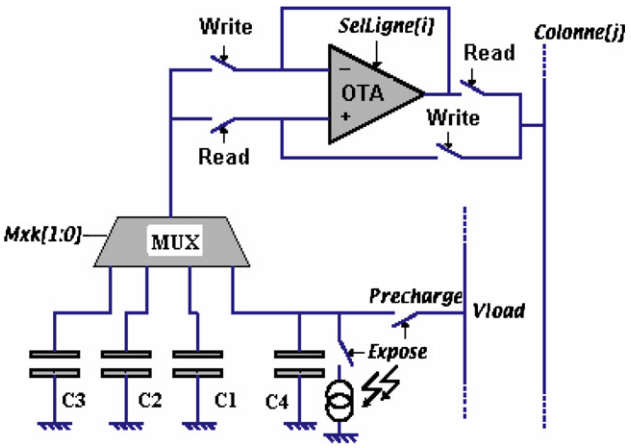


Figure 2. Pixel scheme (OTA: operational transconductance amplifier).

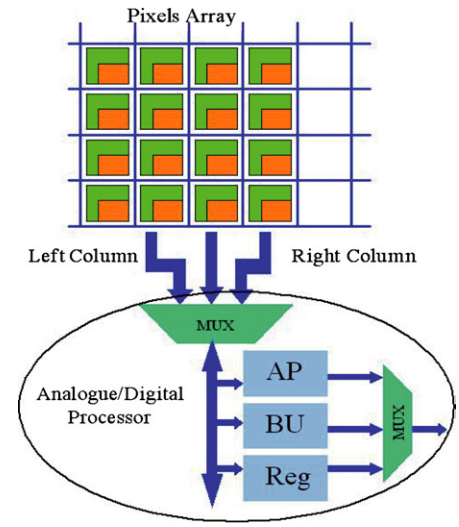


Figure 3. Analogue–digital processor (processing unit) architecture.

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 relative to the length of the columns. Pixels of the same column exchange their data with the corresponding processing element (PE) through a digital analogue bus (DAB). To access any of its four memories, each pixel includes a bi-directional (4 to 1) multiplexer. A set of switches allows selection of 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.

The array of pixels is associated with a vector of processors operating in an analogue/digital mixed mode (figure 3). In this paper, we shall detail only the analogue 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.

Table 1. Main characteristics of the PARIS circuit.

Circuit area (including pads)	10 mm^2
Resolution (pixels)	16 × 16
Number of APUs	16
Pixel area	50 × 50 μm^2
Area per processing unit	50 × 200 μm^2
Clock frequency	10 MHz
Processing unit power consumption	300 μW
Row (16 pixels) power consumption	100 μW

Its functioning is much like a bit stream DAC: an input voltage sets the initial charges in C_{in1}. The iterative activation of switches 'mean' and/or 'reset' reduces the amount of charge 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 values, the charge in C_{in1} is divided by 2. Iterating the operation N times, this step leads to a charge in C_{in1} of the form

$$Q_{in1} = \langle C_{in1} \cdot V_{in1} \rangle / 2^N. \quad (1)$$

Thanks to the OTA, the remaining charge in the capacitor C_{in1} is transferred to C_{out} when switch 'Add', or 'Sub', is '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].

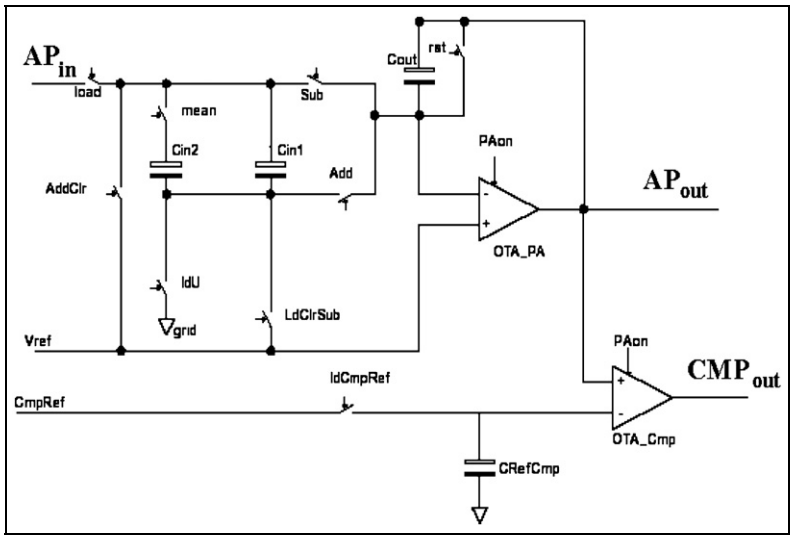
In order to validate this architecture, a first prototype circuit has been designed including 16 × 16 pixels and 16 analogue processing units. This first circuit allows validation of the integrated operators through some image processing algorithms such as edge and movement detection.

Using a standard 0.6 μm CMOS, DLP (double-layer polysilicon) technology, this prototype 'PARIS1' is designed to support up to 256 × 256 pixels. Considering this architecture and the technology used, an artificial retina with higher resolution would lead to hard design constraints on pixel access time and power consumption. To reduce costs the first prototype implements 16 × 16 pixels with 16 analogue processors.

To first order, the accuracy of the computations depends on the dispersion of the component values. The response dispersion between two AP units is 1%. The main characteristics of this chip are summarized in table 1. Note 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 consumption 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 analogue 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 the processing unit power.

3.1.2. Global architecture (PARIS-ARM). We have designed a vision system (PARIS-ARM) based on a PARIS1 retina, implementing converter DAC/ADC and a CPU core: the 16/32-bit ARM7TDMI[1] RISC processor. It is a low-power, general-purpose microprocessor, operating at 50 MHz, developed for custom integrated circuits.

[1] ARM system-on-chip architecture (2nd edition), Steve Furber, September 2000.



AP_{out}: Analog Processor Output
CMP_{out}: Comparator output

AP_{in}: Analog Processor input

Figure 4. Analogue processor (AP) unit scheme.

The embedded in-circuit emulator (ICE) is 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 to 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 analogue processing units, this artificial retina extracts low-level information (e.g. edge 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 terms of computing power, this structure is more advantageous than that based on a CCD sensor associated with a microprocessor [15]. Figure 5 shows the global architecture of the system and figure 6 gives an overview of the experimental module (PARIS-ARM) implemented for the test and measurements.

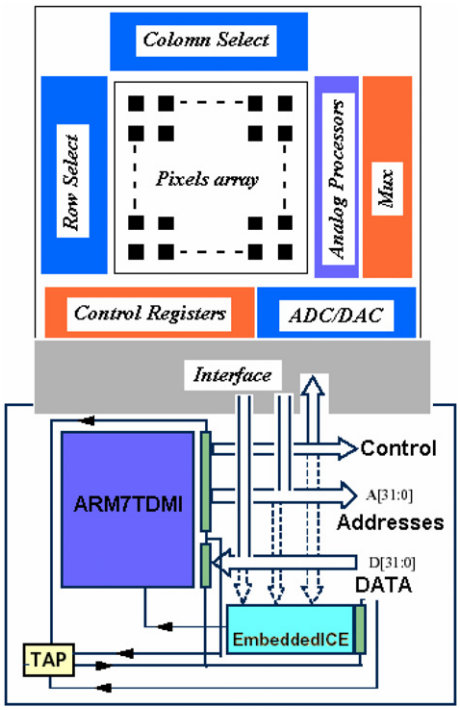


Figure 5. Global architecture of the PARIS1 based vision system.

3.2. A logarithmic CMOS sensor based vision system (FUGA-ARM)

In recent years CMOS image sensors have started to attract 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, the 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, such as, for example, six decades. Image sensors with logarithmic response offer a solution in such situations. However, many works have reported on a 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.

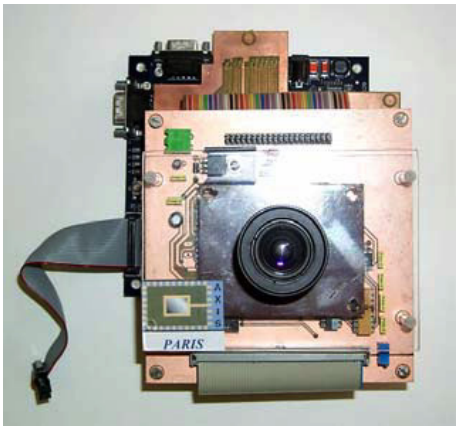


Figure 6. Experimental module PARIS-ARM.

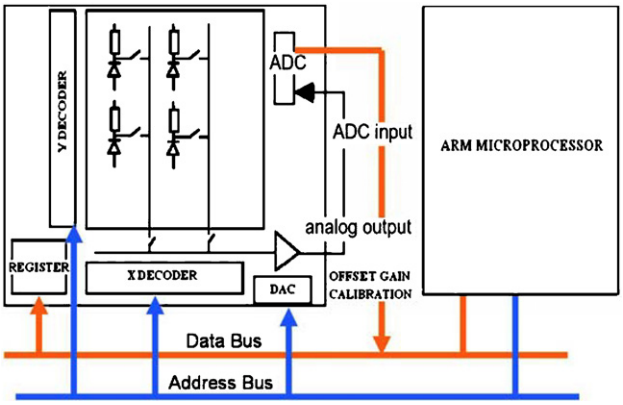
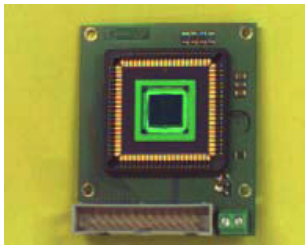


Figure 7. Second architecture implementing a logarithmic CMOS sensor and an ARM7TDMI microprocessor.



Logarithmic CMOS Sensor
(1024x1024 Pixels)

Figure 8. The logarithmic CMOS sensor.

With 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 modelled a vision system (FUGA-ARM) whose architecture, shown in figure 7, is based on a logarithmic CMOS sensor (FUGA1000, figure 8) from FillFactory NV (Belgium) [20] and an ARM microprocessor (the same used for the PARIS-ARM vision system).

The CMOS sensor (FUGA1000) is a 0.4528 inch (type-2/3) random addressable 1024 × 1024 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 the application of an X–Y address, corresponding to the 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.

4. Applications

4.1. On chip image processing

The basis of the smart vision system on chip concept is that analogue 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[2]-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 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 pixel grey-levels at any location by P_{xy}, the response of a mask (3×3 kernel convolution as an example) is given by equations (2) and (3):

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

$$\mathbf{K} = \begin{bmatrix} k_{11} & k_{21} & k_{31} \\ k_{12} & k_{22} & k_{32} \\ k_{13} & k_{23} & k_{33} \end{bmatrix}. \tag{3}$$

The grey-level P_{xy} of the pixel located at the (x, y) position is replaced by the 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 masks (nonlinear filters cannot be implemented in this way).

[2] Digital signal processor.

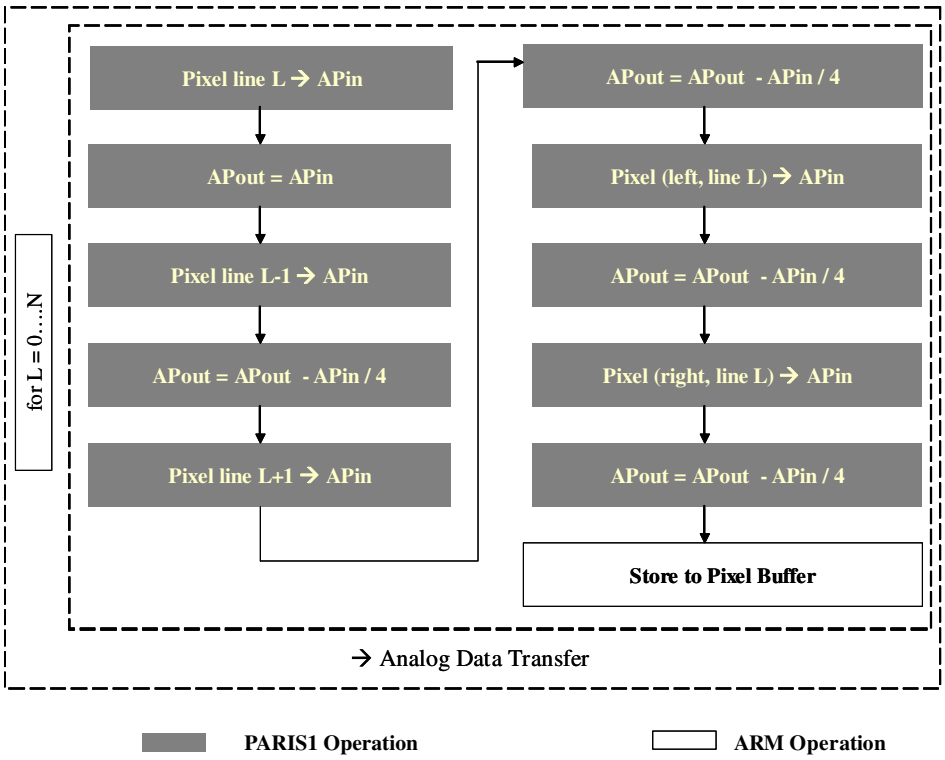


Figure 9. Diagram of the L filter operation.

In the above case, a 3 × 3 local mask was taken as an example for the convolution mask. However, the size of the convolution mask can be expanded to 5 × 5, 7 × 7, 9 × 9, 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 9 allows understanding of the functioning of such an architecture (where each column is assigned to an analogue processor). We choose a classical spatial filter example (a convolution with a 3×3 matrix). The Laplacian kernel L used is given by the matrix (4)

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

Pixels of the same row are simultaneously processed by the analogue processor vector and the computations are iterated on image rows. The arithmetic operations are carried out in analogue. The accumulation of the intermediate results is achieved in the analogue processor using the internal analogue registers. Starting from an acquired image, figure 10 shows the L filtering operation result of an $N \times N$ pixel image, obtained by PARIS1 ($N = 16$). Such an operation is achieved in 6.8 ms. This computation time is global due to $T = N \cdot (T_{add} + 4T_{div} + 4T_{sub})$, where T_{add}, T_{div} and T_{sub} are

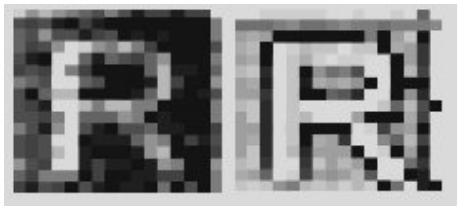


Figure 10. Original image (left) and filtered image (right).

the respective one pixel computation times for an addition, a division and a subtraction operation. The computation time is proportional only to the number of rows and more elaborated algorithms can be implemented similarly.

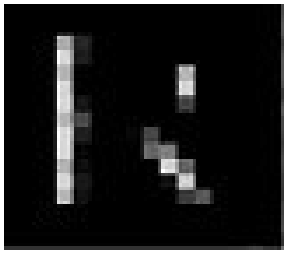
Similar to averaging or smoothing, differentiation can be used to sharpen an image leaving only boundary lines and edges of the objects. This is a high pass filter. The most common methods of differentiation in image processing are the difference, the gradient and Laplacian operators. The difference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

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

Figure 11. Examples of processed images.

detection and estimation. Some examples are presented below. Images are processed with different values of luminosity using the exposure time self-calibration. Figure 11 gives examples of processed images in the luminosity range of 10–1000 Lux.

4.2. Off chip FPN correction and image processing

The major drawback of the logarithmic sensor is the presence of time-invariant noise in the images. The fixed pattern noise (FPN) is caused by the non-uniformity of the transistor characteristics. In particular, threshold voltage variations introduce a voltage-offset characteristic for each pixel. The continuous-time readout of a logarithmic pixel makes 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 from 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 12 shows an image with the FPN and figure 13 gives the 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 2 mV 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.

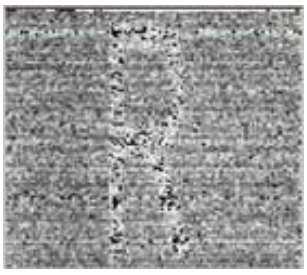


Figure 12. Image with FPN.

Figure 13. Image with FPN removed.

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 the 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

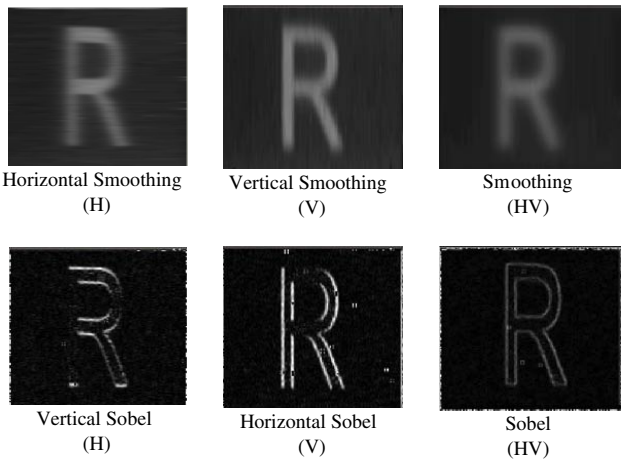


Figure 14. Examples of image processing implemented with the FUGA1000 sensor based vision system.

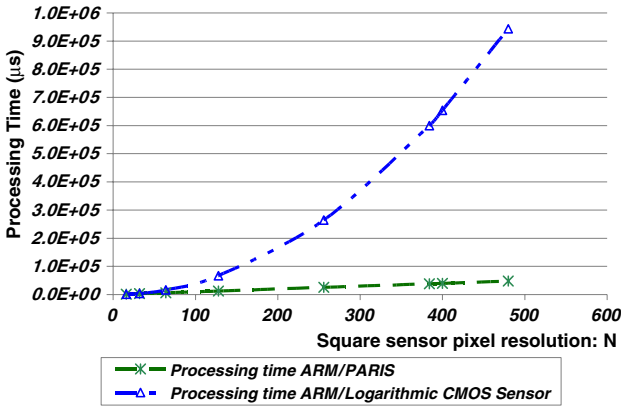


Figure 15. Processing time of an edge detection: PARIS-ARM architecture versus ARM/logarithmic CMOS sensor.

same processor (ARM7TDMI) for the different implemented systems.

The filter we used has been designed by 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. The Garcia Lorca derivator is, after simplification of Deriche, derivator, a 3×3 convolution kernel instead of a recursive derivator,

$$y(n) = (1-\lambda)^2 \cdot x(n) + 2\lambda \cdot y(n-1) - \lambda^2 \cdot y(n-2) \quad (5)$$

$$y(n) = (1-\gamma)^2 x(n) + 2\gamma y(n-1) - \gamma^2 y(n-2), \quad (6)$$

with

$$\gamma = \mathrm{e}^{-\alpha}. \quad (7)$$

Here $x(n)$ 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 14 gives examples of smoothing and derivator filters implemented with the FUGA-ARM vision system and applied to 120×120 pixel images.

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 the PARIS retina (PARIS-ARM). This comparison is related to image processing time and does not 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 pixel saturation and gives an adaptive amplification of pixel output, which is necessary for 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 below. For the artificial retina based system, these computations are carried out by the analogue 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 figure 15 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 pixel number N^2 (quadratic form). Hence, the computation time for the PARIS-ARM system varies according to the row number N (linear form) thanks to the analogue processor vector.

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 row number and C is the column number:

$$\mathrm{CPP} = \frac{T \times F_{CLK}}{L \times C}. \quad (8)$$

Figure 16 shows the evolution of the CPP for the 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 processed pixels. For the PARIS-ARM system, the CPP factor is inversely proportional to the row 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.

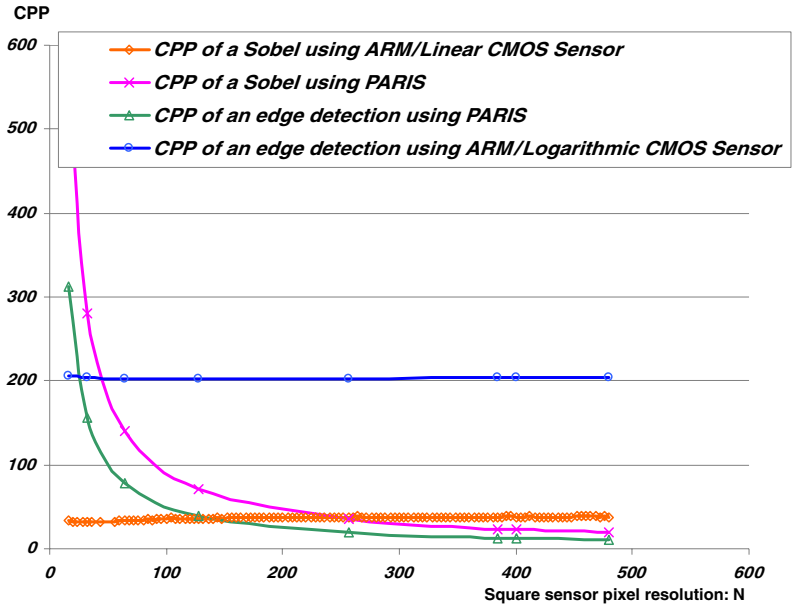


Figure 16. Evolution of the CPP (cycle per pixel) for PARIS-ARM and the FUGA-ARM architectures.

6. Conclusion

When we wish to carry out real-time image acquisition and processing, 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 the benefits of fast and parallel processing. Since each vision algorithm has its own applications and design specifications, it is difficult to predetermine an 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 optimize 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 object tracking.

References

[1] Kogler J, Hemetsberger H, Alefs B, Kubinger W and Travis W 2006 Embedded stereo vision system for intelligent autonomous vehicles *IEEE Intelligent Vehicles Symp. (Tokyo, Japan, 13–15 June)* pp 64–9

[2] Collado J M, Hilario C, Escalera A and Armingol J M 2006 Self-calibration of an on-board stereo-vision system for driver assistance systems *Proc. IEEE Intelligent Vehicles Symp. (Tokyo, Japan, 13–15 June)* pp 156–62

[3] Stoppa D, Simoni A, Gonzo L, Gottardi M and Betta G 2002 A 138 dB dynamic range CMOS image sensor with new pixel architecture *IEEE Int. Solid State Circuits Conf.* pp 40–1

[4] El Gamal A *et al* 1999 Pixel level processing: why, what and how? *Proc. SPIE* **3650** 2–13

[5] Ni Y and Guan J H 2000 A 256 × 256-pixel smart CMOS image sensor for line based stereo vision applications *IEEE J. Solid-State Circuits* **35** 1055–61

[6] Moini A 2000 Vision chips or seeing silicon *Technical Report* Centre for High Performance Integrated Technologies and Systems, The University of Adelaide, March 1997 (Dordrecht: Kluwer Academic)

[7] Chen S, Bermak A, Wan W and Martinez D 2006 A CMOS image sensor with combined adaptive-quantization and QTD-based on-chip compression processor *IEEE Custom Integrated Circuits Conf. (San Jose, CA, Sept.)*

[8] Dudek P 2000 A programmable focal-plane analogue processor array *PhD Thesis* University of Manchester Institute of Science and Technology (UMIST) (May 2000)

[9] Arias-Estrada M 2001 A real-time FPGA architecture for computer vision *J. Electron. Imaging (SPIE—IS&T)* **10** 289–96

[10] Zhou Z, Pain B and Fossum E 1997 Frame-transfer CMOS active pixel sensor with pixel binning *IEEE Trans. Electron Devices* **44** 1764–8

[11] Seguine D 2002 Just add sensor—integrating analogue and digital signal conditioning in a programmable system on chip *Proc. IEEE Sensors* **1** 665–8

[12] Dupret A, Klein J O and Nshare A 2002 A DSP-like analogue processing unit for smart image sensors *Int. J. Circuit Theory Appl.* **30** 595–609

[13] Dupret A, Belhaire E and Rodier J-C 1996 A high current large bandwidth photosensor on standard CMOS process *EuroOpto'96, AFPAEC (Berlin)*

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

[15] Litwiller D 2001 CCD vs. CMOS: facts and fiction *Photonics Spectra* (January) 151–4

[16] Litwiller D 2005 CCD vs. CMOS: maturing technologies, maturing markets *Photonics Spectra* (August) 54–8

[17] Dierickx B and Bogaerts J 2004 Advanced developments in CMOS imaging *Fraunhofer IMS workshop (Duisburg, 25 May)*

[18] Loose M, Meier K and Schemmel J 1998 CMOS image sensor with logarithmic response and self calibrating fixed pattern noise correction *Proc. SPIE* **3410** 117–27
[19] Choubey B *et al* 2006 An electronic-calibration scheme for logarithmic CMOS pixels *IEEE Sensors J.* **6** no 4
[20] FUGA1000-EVB from FillFactory: http://www.framos.co.uk/news/newsletter/Notes0301.pdf
[21] Bovik A (ed) 2000 *Handbook of Image & Video Processing* (San Diego, CA: Academic)
[22] Blanchet G and Charbit M 2005 *Digital Signal and Image Processing Using Matlab* (London: Iste)
[23] Deriche R 1990 Fast algorithms for low level-vision *IEEE Trans. Pattern Anal. Mach. Intell.* **12** 78–88
[24] Garcia Lorca F, Kessal L and Demigny D 1997 Efficient ASIC and FPGA implementation of IIR filters for real time edge detections *Proc. Int. Conf. on Image Processing (IEEE ICIP)*
[25] Elouardi A, Bouaziz S, Dupret A, Lacassagne L, Klein J O and Reynaud R 2006 A smart sensor for image processing: towards a system on chip *IEEE Int. Symp. on Industrial Electronics (Montréal, Canada, 9–12 July)*