

# Image Processing: Towards a System on Chip

A. Elouardi, S. Bouaziz, A. Dupret, L. Lacassagne,  
J.O. Klein and R. Reynaud  
*Institute of Fundamental Electronics, Paris XI University, Orsay  
France*

## 1. Introduction

Many kinds of vision systems are available on today's market with various applications. Despite the wide variety of these applications, all digital cameras have the same basic functional components, which consist in photons collection, wavelength photons discrimination (filters), timing, control and drive electronics for the sensing elements, sample/hold operators, colours processing circuits, analogue to digital conversion and electronics interfaces (Fossum, 1997).

Today, robotics and intelligent vehicles need sensors with fast response time, low energy consumption, able to extract high-level information from the environment (Muramatsu et al., 2002). Adding hardware computation operators near the sensor increases the computations potentiality and reduces inputs/outputs operations towards the central processor unit.

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

However, in the last decade, CMOS image 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 process.

Still, raw output images acquired by CMOS sensors present poor quality for display and 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.

To face the high data flow induced by the computer vision algorithms, an alternative approach consists in performing some image processing on the sensor focal plane. 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 some simple low-level image processing tasks (early-vision), a smart sensor integrates analogue and/or digital processing circuits in the pixel (Burns et al., 2003, El Gamal et al., 1999, Dudek, Hicks, 2000) or at the edge of the pixels array (Ni, Guan, 2000). Most often, such circuits are dedicated for specific applications. The energy dissipation is weak compared to that of the traditional approaches using multi chip (microprocessor,

Source: Image Processing, Book edited by: Yung-Sheng Chen,  
ISBN 978-953-307-026-1, pp. 572, December 2009, INTECH, Croatia, downloaded from SCIYO.COM

sensor, logic glue ...etc) (Alireza, 2000). Noise and cross-talk can also be reduced through monolithic connections instead of off-chip wires.

Moreover, this chapter is built to get a conclusion on the aptitude of the retinas to become potential candidates for systems on chip, consequently to reach an algorithm-architecture and system adequacy. In this context, an application was selected making it possible to develop a conclusion on a partial integration of a system on chip. Hence this chapter focuses on the VLSI compatibility of retinas, more particularly, of integrating image processing algorithms and their processors on the same sensor focal plane to provide a smart on chip vision system (System on Chip). It discusses why the retina is advantageous, what elementary functions and/or operators should be added on chip and how to integrate image processing algorithms (i.e. how to implement the smart sensors). The chapter includes recommendations on system-level architectures, applications and discusses the limitations of the implementation of smart retinas, which are categorized by the nature of image processing algorithms, trying to answer the following questions:

- Why vision algorithms (image processing algorithms) should be implemented by the retinas?
- What algorithms and processing components should be put with retinas to provide a part or a whole system on chip?
- How to aggregate these processing operators (by pixel, by group of pixels, by column, by line or for the whole array)?
- What structures are the best suited for each class of image processing algorithms?

To sustain the discussion, we propose a system-level architecture and a design methodology for integrating image processing within a CMOS retina on a single chip. It highlights a compromise between versatility, parallelism, processing speed and resolution. Our solution aims to take also into account the algorithms response times, the significant resolution of the sensor, while reducing energy consumption for embedding reasons so as to increase the system performances.

We have done a comparison relating two different architectures dedicated for a vision system on chip. The first one implements a logarithmic APS imager and a microprocessor. The second involves the same processor with a CMOS retina that implements hardware operators and analogue microprocessors. We have modeled two vision systems. The comparison is related to image processing speed, processing reliability, programmability, precision, subsequent stages of computations and power consumption.

## 2. Systems description

### 2.1 On chip vision system: why smart retinas?

The smart retinas focus on analogue VLSI implementations even though hardware implementation of image processing algorithms typically refers to digital implementations. The main interest is to adjust the functionality and the quality of the processing. Compared to a vision processing system consisting of a combination of a CMOS imagers and a processor in separate chips, a smart retina provides many advantages:

- Processing speed: the information transfer occurs serially between the imager and the associated processor, while in smart sensor data can be processed and transferred in parallel. Consequently, the processing speed can be enhanced: parallel operations between image acquisition and processing occur without digital sampling and quantization.

- **Single chip integration:** a single chip implementation of smart sensors contains image acquisition, low and high-level image processing circuits. A tiny sized chip can do the equivalent work of a camera associated to a computer or a DSP.
- **Adaptation:** Conventional cameras have at best an automatic gain control with offset tuning at the end of the output data channel. In smart sensors, photodetectors and operators are co-located in the pixel for a local or global adaptation that enhances their dynamic range.
- **Power dissipation:** a large portion of the total power is due to off-chip connections. On-chip integration reduces power consumption.
- **Size and Cost:** Analogue implementations of image processing algorithms feature a more compact area than their digital counter part. This is a crucial design issue for smart sensors. While a simple computation of large digital bit consumes a large area for the component design, a simple analogue component with compact size can typically compute the equivalent operation. The single chip implementation of the sensor and the processor can reduce the system size. The compact size of the chip is directly related to the fabrication cost.

Although designing single chip sensors is an attractive idea and the integration of image sensing and analogue processing has proven to be very striking, it faces several limitations well described and well argued in (Alireza, 2000):

- **Processing reliability:** Processing circuits of smart sensors often use unconventional analogue circuits which are not well characterized in many current technologies. As a result, if the smart sensor does not take in account the inaccuracies, the processing reliability is severely affected.
- **Custom designs:** Unconventional analogue or digital operators are cells often used in implementation of smart sensors. Operators from a design library cannot be used, and many new schemes and layout have to be developed. Their design can take a long time and the probability of design errors is higher.
- **Programmability:** most smart sensors are not general-purpose devices, and are typically not programmable to perform different vision. This lack of programmability is undesirable especially during the development of a vision system when various simulations are required.

Even with these disadvantages, smart sensors are still attractive, mainly because of their effective cost, size and speed with various on-chip functionalities (Rowe, 2001, Seguire, 2002). Simply, benefits exist when a camera with a computer system are converted into a small sized vision system on chip (SoC).

## **2.2 Proof-of-concept: a retina based vision system**

### **2.2.1 On-chip image processing: review of integrated operators on smart circuits**

Many vision algorithms of on-chip image processing with CMOS image sensors have been developed (Koch, 1995, Kleinfelder, 2001): image enhancement, segmentation, feature extraction and pattern recognition. These algorithms are frequently used in software-based operations, where structural implementation in hardware is not considered. Here, the main research interest focuses on how to integrate image processing (vision) algorithms with CMOS integrated systems or how to implement smart retinas in hardware, in terms of their system-level architectures and design methodologies.

Different partitions for the architectural implementation of on-chip image processing with CMOS image sensors are proposed. The partition does not only take in account 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 sensors. 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 consists of one processing element (PE) per image sensor 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 high SNR, 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 PE like the approach presented by P. Dudeck in (Dudek, 2000).

In a view of great block partitioning, a global processing unit can be instantiated, beside the array of sensors, from a library. 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 high complexity of design and the high power consumption of the chip (Arias-Estrada, 2001).

Another implementation structure is the frame memory processing. A memory array with the same number of elements as the sensor is located below the imager array. Typically, the image memory is an analogue frame memory that requires less complexity of design, area, and processing time (Zhou, 1997). 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.

### 2.2.2 PARIS architecture

PARIS (Parallel Analogue Retina-like Image Sensor) is an architecture for which the concept of retinas is modeled implementing in the same circuit an array of pixels, integrating memories, and column-level analogue processors (Dupret, 2002). The proposed structure is shown in figure 1. This architecture 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 approach presents the advantage to enable the design of complex processing units without decreasing the resolution. In return, because the parallelism is reduced to a row, 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 image sensor array, the column processing offers the advantages of parallel processing that permits low frequency and thus low power consumption. Furthermore, it becomes possible to chain basic functions in an arbitrary order, as in any digital SIMD machine. The resulting low-level information extracted by the retina can be then processed by a digital microprocessor.

The array of pixels constitutes the core of the architecture. Pixels can be randomly accessed. 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 (Dupret, 1996) 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  $V_{ref}$ . In some cases, the semi-parallel processing imposes to store intermediate and temporary results for every pixel in four MOS capacitors used as analogue memories (figure 2). One of the four memories is used to store the analogue voltage deriving from the sensor. The pixel area is  $50 \times 50 \mu\text{m}^2$  with a Fill Factor equal to 11%.

This approach eliminates the input/output bottleneck between different 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 implementation of the processing is more flexible relatively to the length of the columns. Pixels of the same column exchange their data with the corresponding processing element through a Digital Analogue Bus (DAB). So as to access any of its four memories, each pixel includes a bidirectional (4 to 1) multiplexer. A set of switches makes possible to select the voltage stored in one of 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.

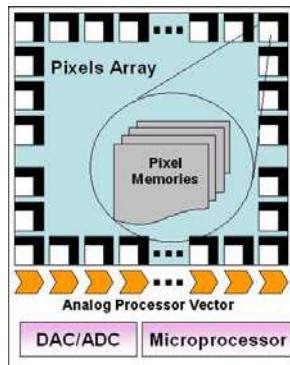


Fig. 1. PARIS architecture

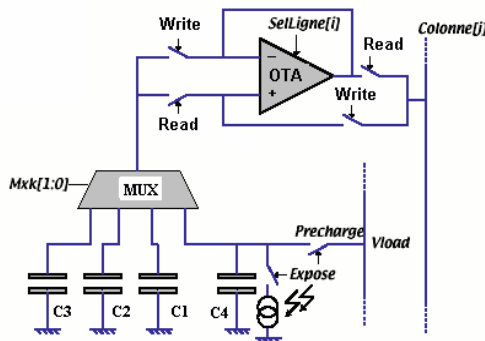
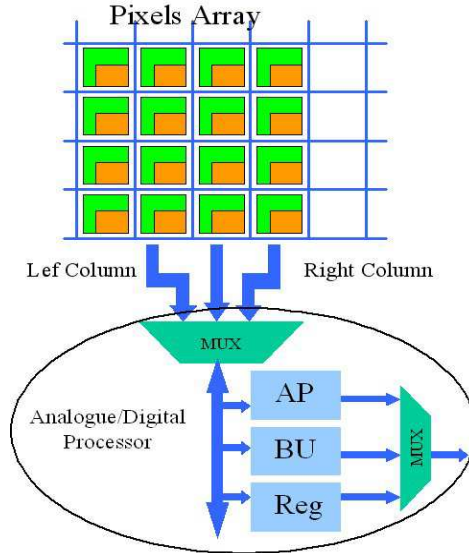


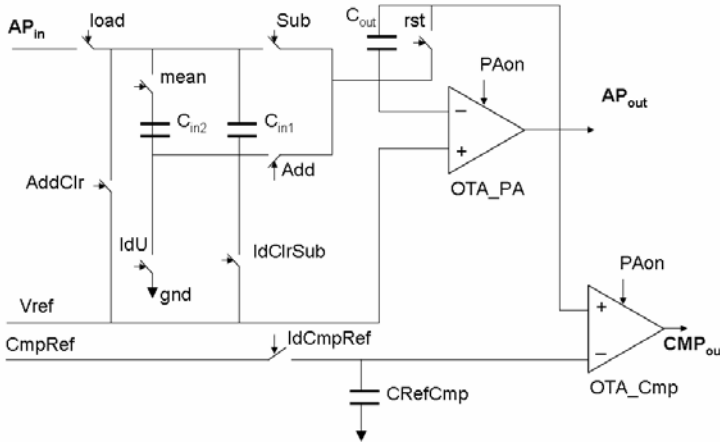
Fig. 2. Pixel scheme

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



**AP:** Analogue Processor    **Reg:** Registers  
**BU:** Boolean Unit        **Mux:** Multiplexer

Fig. 3. Analogue processor interface



**AP<sub>out</sub>:** Analogue Processor output    **AP<sub>in</sub>:** Analogue Processor input  
**CMP<sub>out</sub>:** Comparator output

Fig. 4. Analogue-digital processor unit

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 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 given by the equation (1):

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

Thanks to the OTA, the remaining charge in capacitor  $C_{in1}$  is arithmetically 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 (Dupret, 2000).

In order to validate this architecture, a first prototype circuit has been designed including  $16 \times 16$  pixels and 16 analogue processing units. This first circuit allows validating the integrated operators through some image processing algorithms. Using a standard  $0.6 \mu\text{m}$  CMOS, DLM-DLP technology, this prototype “PARIS1” is designed to support up to  $256 \times 256$  pixels. Considering this architecture and the technology used, higher resolution retina would lead to hard design constraints such as pixel access time and power consumption. As to reduce costs the prototype implements  $16 \times 16$  pixels with 16 analogue processors. Yet, this first circuit allows validating the integrated operators through some image processing algorithms like edge and movement detection. At a first order, the accuracy of the computations depends on the dispersion of the components values. The response dispersion between two APE units is 1%. A microphotography and a view of a first prototype of PARIS circuit are given in figure 5. The main characteristics of this vision chip are summarized in Table 1. Notice that the given pixel power consumption is its peak power i.e. when pixel is addressed. In other cases the OTA of the pixels are switched off and the pixel power consumption is only due to C4 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:  $C \cdot (P_{\text{pixel}} + P_{\text{Processing Unit}})$ , where  $C$  is the chip number of columns.

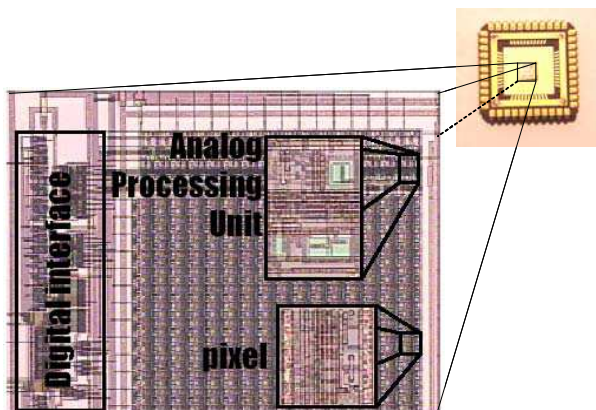


Fig. 5. Microphotography and a  $16 \times 16$  pixels prototype of PARIS sensor

Circuit area (including pads)	10 mm <sup>2</sup>
Resolution (Pixels)	16x16
Number of APUs	16
Pixel Area	50x50 μm <sup>2</sup>
Area per Processing Unit	50x200 μm <sup>2</sup>
Clock Frequency	10 MHz
Processing Unit Power Consumption	300 μW
16 Pixels Line Power Consumption	100 μW

Table 1. Main characteristics of PARIS circuit

A finer analysis of the circuit performance (figure 6) shows that the time allocated to analogue operations is considerable. This problem can be solved in two ways. Either we increase the number of input in the analogue processor, or we give the opportunity to perform multiplications on a single clock (Moutault, 2000).

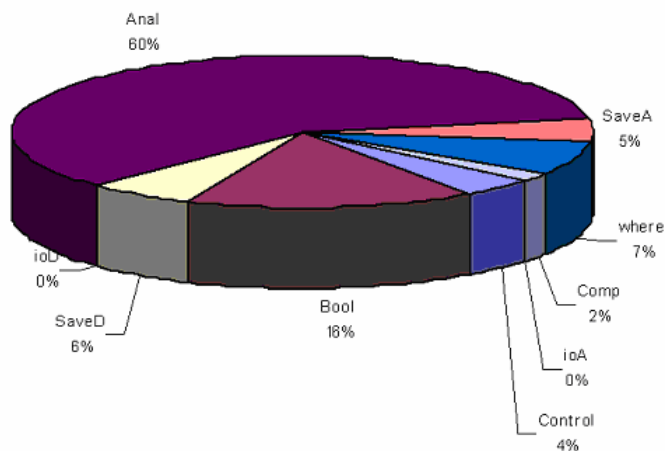


Fig. 6. Instructions occurrences based on multiples tests

### 2.2.3 Global architecture

To evaluate an on chip vision system architecture, we have implemented a vision system based on PARIS retina, implementing DAC/ADC converter and a CPU core: the 16/32-bit ARM7TDMI1 RISC processor. It is a low-power, general purpose microprocessor, operating at 50 MHz, developed for custom integrated circuits.

The Embedded ICE logic 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. The processor waits for the extracted low-level

<sup>1</sup> ARM System-on-Chip Architecture (2<sup>nd</sup> Edition), Steve Furber, September 2000.



information and processes them to give high-level information. The system sends then sequences of entire raw images.

With all components listed above, we obtain a system vision that uses a fully programmable smart retina. Thanks to the analogue processing units, this 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 type remains in the parallel execution of a consequent number of low level operations in the array by 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 (Litwiller, 2001). Figure 7 shows the global architecture of the system and figure 8 gives an overview of the experimental module implemented for test and measurements.

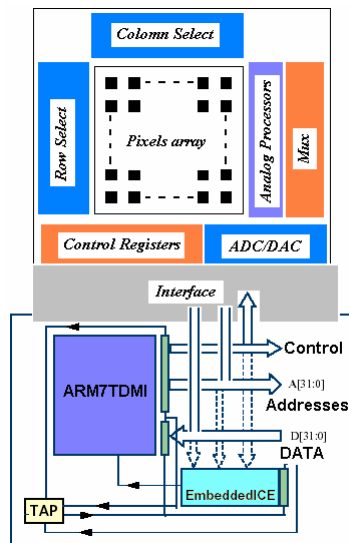


Fig. 7. Global architecture



Fig. 8. Experimental module overview

### 2.3 Proof-of-concept: a vision system based on a logarithmic CMOS sensor

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 imaging 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 (Dierickx, 2004, Walschap, 2003). An example 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 (Loose, 1998) have been reported on high dynamic range CMOS sensor having a 130dB dynamic. These sensors may be the alternative to logarithmic CMOS sensors.

Since the sensor is a non-integrating sensor there is no control of the integration time. Because of the large logarithmic response the sensor can deal with images with large contrast without the need for iris control, simplifying the system vision. This makes this sensors very well suited for outdoor applications.

Due to the random access, regions of interest can to be read-out and processed. This reduces the image processing, resulting in faster and/or cheaper image processing systems.

We have modeled a vision system based on a logarithmic CMOS sensor (FUGA1000) (Ogiers, 2002) and an ARM microprocessor (the same used for the first vision system based on PARIS retina). The entire architecture is shown in figure 9. Figures 10.a and 10.b gives an overview of the CMOS sensor and the experimental module.

The CMOS sensor (FUGA1000) is an 11.5 mm (type-2/3") random addressable 1024 x 1024 pixels. It has a logarithmic light power to signal conversion. This monolithic digital camera chip has on-chip a 10 bit flash 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 matrix, a 10 bit digital word corresponding to light intensity on the addressed pixel is returned.

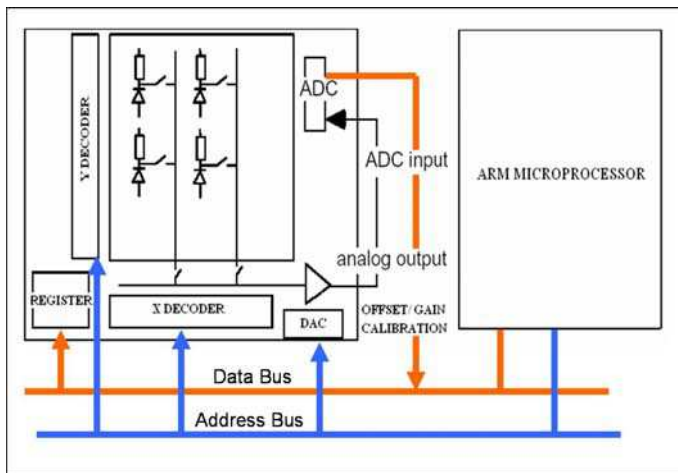


Fig. 9. Second architecture implementing a logarithmic CMOS sensor and an ARM7TDMI microprocessor

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) is not possible.

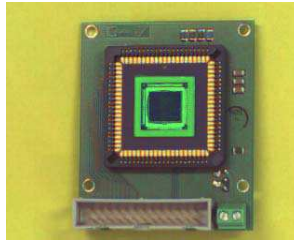


Fig. 10.a. Logarithmic CMOS sensor (1024x1024 pixels)

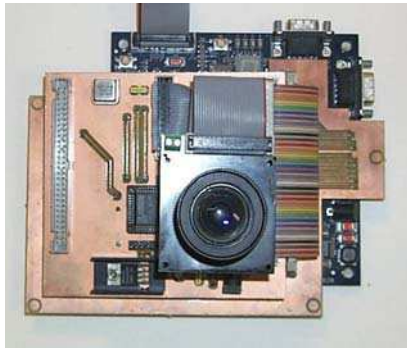


Fig. 10.b. Instrumental module overview with the CMOS sensor

### 3. Applications

#### 3.1 Exposure time calibration algorithm

Machine vision requires an image sensor able to capture natural scenes that may have a dynamic adaptation for intensity. Reported wide image sensors suffer from some or all of the following problems: large silicon area, high cost, low spatial resolution, small dynamic range, poor pixel sensitivity, etc.

The primary focus of this research is to develop a single-chip imager for machine vision applications which resolves these problems, able to provide an on-chip automatic exposure time algorithm by implementing a novel self exposure time control operator. The secondary focus of the research is to make the imager programmable, so that its performance (light intensity, dynamic range, spatial resolution, frame rate, etc.) can be customized to suit a particular machine vision application.

Exposure time is an important parameter to control image contrast. This is the motivation for our development of a continuous auto-calibration algorithm that can manage this state for our vision system. This avoids pixels saturation and gives an adaptive amplification of the image, which is necessary to the post-processing.

The calibration concept is based on the fact that since the photo-sensors are used 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 decrease, on the other hand, if the luminosity is low the exposure time should increase. Hence lower is the exposure time

simpler is the image processing algorithms. This globally will decrease response time and simplify algorithms. We took several measurements with our vision system, so that we can build an automatic exposure time checking algorithm according to the scene luminosity. Figure 11 presents the variation of the maximum grey-level according to the exposure time. For each curve, we note a linear zone and a saturation zone. Thus we deduce the gradient variation ( $\Delta_{max}/\Delta t$ ) according to the luminosity. The final curve can be scored out as a linear function (figure 12).

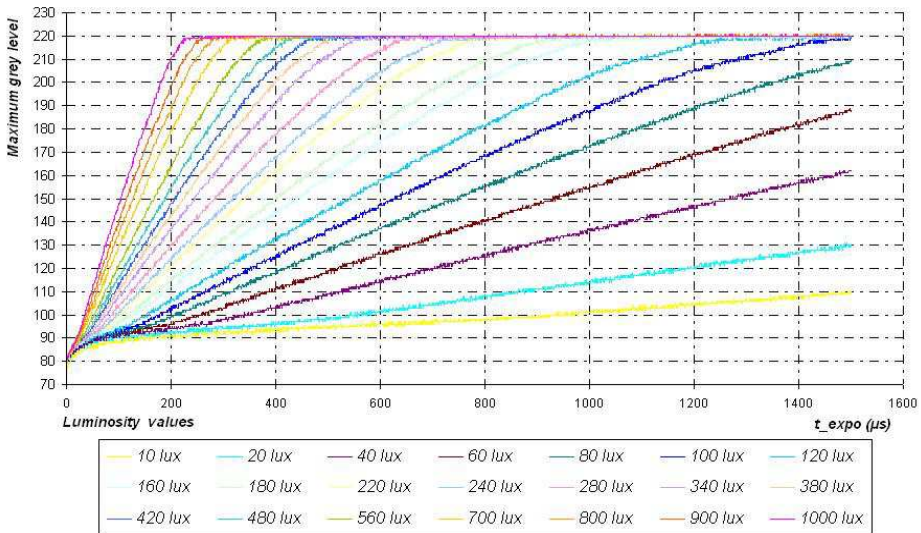


Fig. 11. Measured results (Maximum grey-level versus exposure time for different values of luminosity)

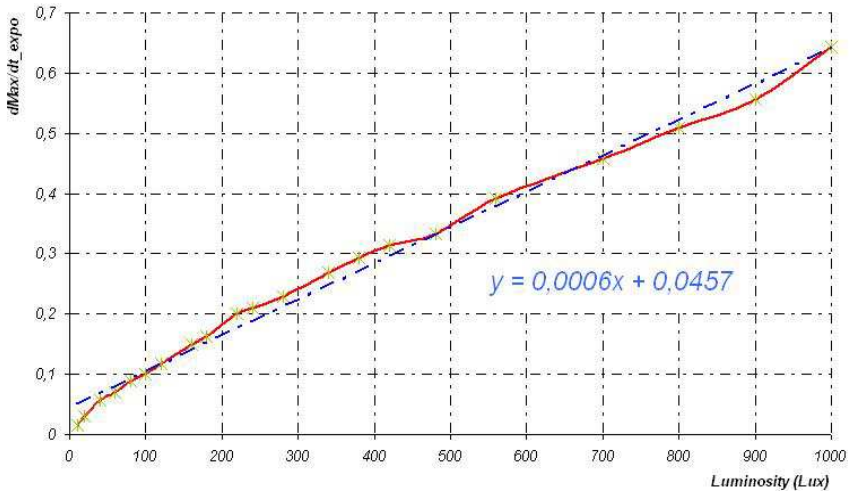


Fig. 12. Gradient variation according to the luminosity

The algorithm consists in keeping the exposure time in the interval where all variations are linear and the exposure time is minimal. Control is then initialised by an exposure time belonging to this interval. When a maximum grey-level is measured, the corresponding luminosity is deduced and returns a gradient value which represents the corresponding slope of the linear function. Figure 13 gives an example of images showing the adaptation of the exposure time to the luminosity.

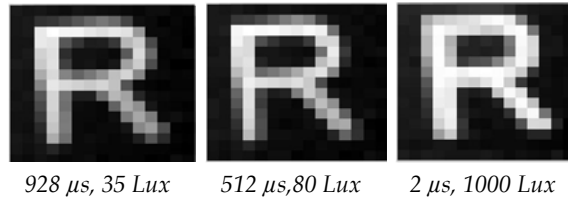


Fig. 13. Exposure time adaptation to the luminosity

### 3.2 On Chip image processing

Yet, in this chapter, we do not wish to limit implementations to application-specific tasks, but to allow for general-purpose applications such as DSP-like image processors with programmability. The idea is based on the fact that some of early level image processing, in the general-purpose chips, is commonly shared with many image processors, which do not require programmability on their operation.

These early level image processing algorithms, from the point of views of on-chip implementation, are relatively pre-determined and fixed, where their low precision can be compensated later by back-end processing. Here, we will investigate what 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. The order of this image analysis can vary for different applications, and stages of the processes can be omitted. In image processing, the image acquisition is used to capture raw images from its input scene, through the use of video camera, scanners and, in the case of smart retinas, the solid-state arrays.

Local operation is also called mask operation where each pixel is modified according to the values of the pixel's neighbors (typically using convolution masks). In aspects of on-chip integration with image sensors, these operations provide advantages of real time process in image acquisition and processing, such as implementations of many practical linear spatial image filters and image enhancement algorithms. In addition, because the local operation is feasible for column structure implementations, low frequency processing is enabled and thus low power consumption is possible. However, since the local operations are based on a technique where local memory stores pixel values of the neighbors and processes them concurrently, implementation of the operation must contain some type of storage. Applications of local operations typically use an iterative technique for advanced image enhancement algorithms, which cannot practically be implemented on-chip. Nevertheless, in the case of column structure implementations, local operation still has a limitation on design area because of the restricted column width, even with flexible design area in the

vertical direction. Therefore, in order to overcome these limitations, careful designs and system plans are required for the on-chip implementations.

In order to understand the nature of a local operation and to find an adequation relationship between algorithms and on chip architectural implementations, we will look into the main algorithms, grouped according to the similarity of functional processing. The diagram presented in figure 14 allows understanding the functioning of such architecture (where each column is assigned to an analogue processor). We chose a traditional example consisting of a spatial filter which is a 3x3 convolution kernel K, implementing a 4-connex laplacian filter. The convolution kernel K used is given by the table (2):

0	-1/4	0
-1/4	1	-1/4
0	-1/4	0

Table 2. Convolution kernel

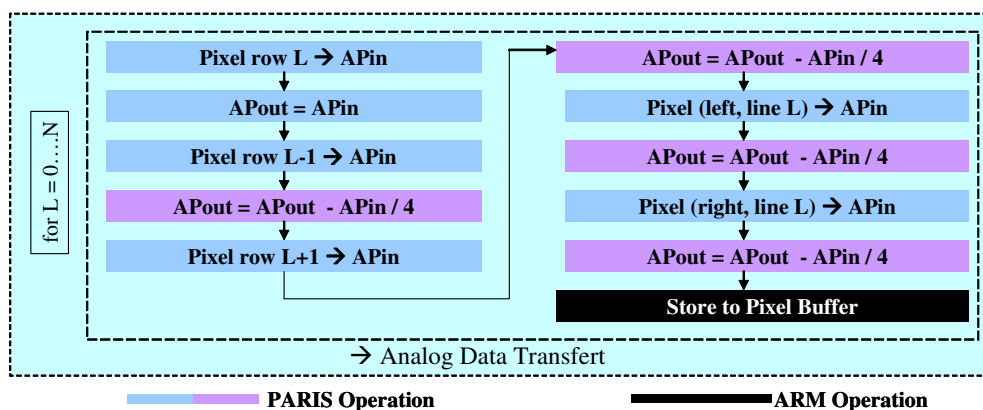


Fig. 14. Diagram of the K filter operation



Fig. 15. Original image (left) and filtered image (right)

Pixels of the same line are simultaneously processed by the analogue processor (AP) vector and the computing is iterated on image rows. The arithmetic operations (division, addition) are carried out in analogue. The accumulation of the intermediate results is achieved in the analogue processor by using the internal analogue registers. Starting from an acquired image, the figure 15 shows the K filtering operation result of an NxN pixels image, obtained by PARIS1 when N=16. Such operation is achieved in 6887  $\mu$ s. This computation time is globally due to:

$T = N \cdot (T_{\text{add}} + 4 \cdot T_{\text{div}} + 4 \cdot T_{\text{sub}})$  where  $T_{\text{add}}$ ,  $T_{\text{div}}$  and  $T_{\text{sub}}$  are the computation time, for one pixel, of the addition, division and subtraction operation. Of course, the computation time is proportional only to the number of rows in the sensor and more elaborated algorithms can be implemented similarly.

This operation can be carried out using the four analogue memories integrated in each pixel: for each subtraction and division and for each neighbour pixel, the result can be stored in one of the reports memories plans. Each memory can store an intermediate result. The final result can be obtained finally by a simple addition or subtraction achieved by the analogue processor vector. We obtain the filtered image by iterating on all array's rows. Such operation is processed in 6833  $\mu\text{s}$ . This computation time is globally due to:

$$T = N \cdot (T_{\text{add}} + 4 \cdot T_{\text{sub}}) + N \cdot T_{\text{div}}$$

This second method reduces the computing time and it is significant when the number of rows grows. Here for our example, it enabled us to reduce the computing time of 50  $\mu\text{s}$  for 16x16 pixels image. Saved time will be of 0.8ms for an image of 256x256 pixels. The control and addressing of the PARIS retina requires more ARM program computing resources to establish an FSM (Finite State Machine). PARIS retina can accept more control and addressing flow than what it is sent by the ARM programmed FSM controller. Hardware FSM version can deliver more control flow. So, our experimental results give low limit bandwidth of the retina control flow.

Opposite to integration that is similar to averaging or smoothing, differentiation can be used to sharpen an image leaving only boundary lines and edges of the objects. This is an extreme case of high pass filters. The most common methods of differentiation in image processing applications are first and second derivatives: gradient and laplacian operators. The difference filter is the simplest form of the differentiation with subtracting adjacent pixels from the centred pixel in different directions. The gradient filters represent the gradients of the neighbouring pixels (image differentiation) in forms of matrices. Such gradient approaches and their mask implementations are represented with various methods: Roberts, Prewitt, Sobel, Kirsch and Robinson.

With many different local operations in image processing algorithms, these operations can be categorized into three major groups: smoothing filters, sharpening filters and edge detection filters. Examples of the local operation algorithms are described in (Bovik, 2000).

We have successfully implemented and tested a number of algorithms, including convolution, linear filtering, edge detection, segmentation, motion detection and estimation. Some examples are presented below. Images are processed at different values of luminosity [60 Lux, 1000 Lux] using the exposure time self calibration. Figure 16 gives examples of processed images using the exposure time calibration algorithm.

### 3.3 Calibration of the CMOS sensor and off-chip 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.

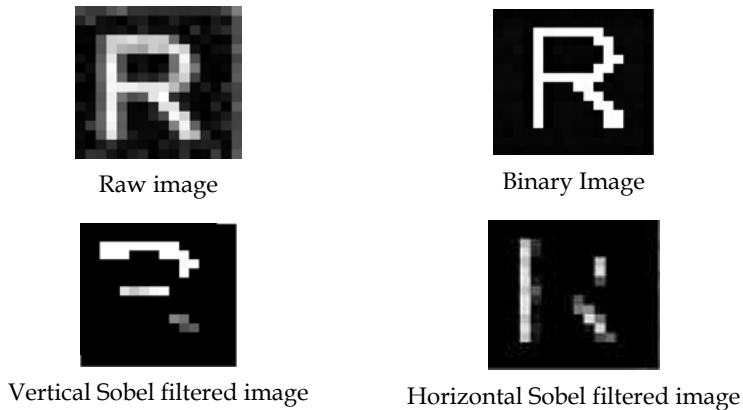


Fig. 16. Examples of image processing

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 adding to each pixel value the corresponding offset.

For the CMOS/APS sensor, the FPN suppression 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 17 shows an image with the FPN and the image after the 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 about 0.2 mV RMS.

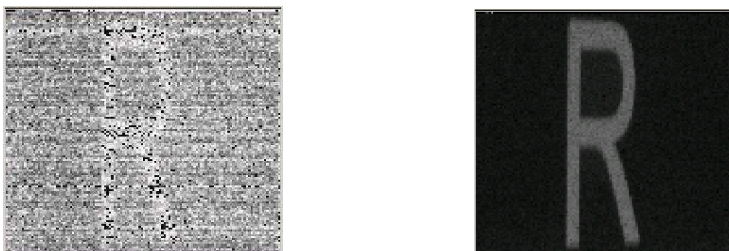


Fig. 17. Images with FPN (left) and with removed FPN (right)

For the FUGA1000 sensor based vision system, images are processed on the ARM microprocessor. We established several algorithms of image processing similar to those established for PARIS based vision system. Other more complicated algorithms which require diversified computing with exponential power were also established. 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 (Deriche, 1990). This filter is a simplification of the Deriche filter (Garcia Lorca, 1997), 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.

$$y(n) = (1 - \lambda)^2 x(n) + 2\lambda y(n-1) - \lambda^2 y(n-2) \quad \text{with} \quad \lambda = e^{-\alpha} \quad (2)$$

$X(n)$  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.  $\lambda$  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  $\alpha$ :  $\alpha=[0.8,1.0]$ . Figure 18 gives examples of smoothing filter and derivator filter implemented with the FUGA-ARM vision system and applied to 120x120 pixels images.

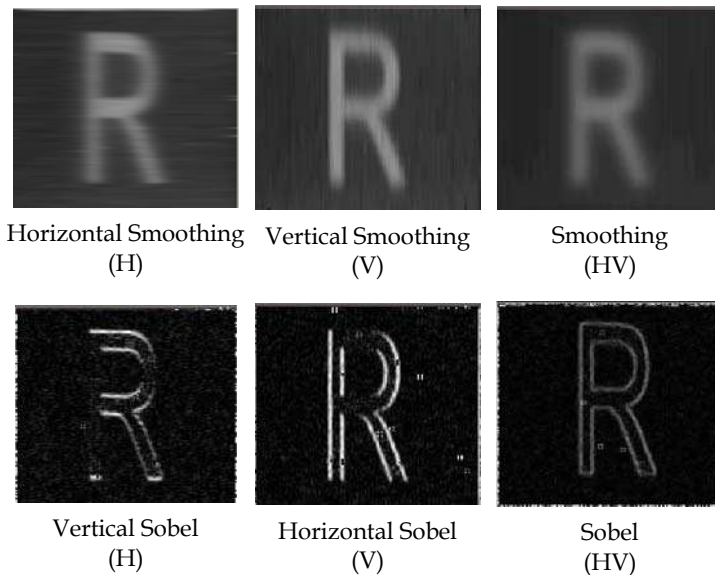


Fig. 18. Examples of image processing implemented with the FUGA1000 sensor based vision system

#### 4. Comparison: standard CMOS sensors versus retina

The aim is to compare the vision system implementing the logarithmic CMOS imager (FUGA1000) and the ARM microprocessor with the one based on PARIS retina (see section B.2). This comparison is related to image processing speed, programmability and subsequent stages of computations.

We have used the 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 retina based system, these computations are carried out by the analogue processors

integrated on-chip. For the FUGA1000 sensor based system, these computations are carried out by the ARM microprocessor.

The two computation time graphics presented in the figure 19 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 retina were obtained by extension of the data processing timing obtained from those of the first prototype (Dupret, 2002). Figure 20 presents the same kind of comparison between PARIS system and a third commercial camera system: EtheCam (Neuricam, Italy). This camera is based on a linear CMOS sensor and an ARM7TDMI microprocessor.

We deduce that the computation time for the FUGA1000 like system varies according to the pixels number  $N^2$  (quadratic form). Hence, the computation time for Retina like system varies according to the number of line  $N$  (linear form) thanks to the analogue processor vector.

Consequently, the microprocessor of the FUGA1000 like system carries out a uniform CPP (Cycle Per Pixel) relative to regular image processing independently of the number of proceeded pixels. For PARIS like system, the CPP factor is inversely proportional to the number of lines  $N$ . Figure 21 shows the evolution of the CPP for PARIS and FUGA1000/ARM systems.

A characterization of the power consumption for PARIS based system has been achieved (Dupret, 2002). The total power of an  $N \times N$  resolution and  $N$  analogue processing units is:

$$P = 100.N^2 + 300.N \tag{3}$$

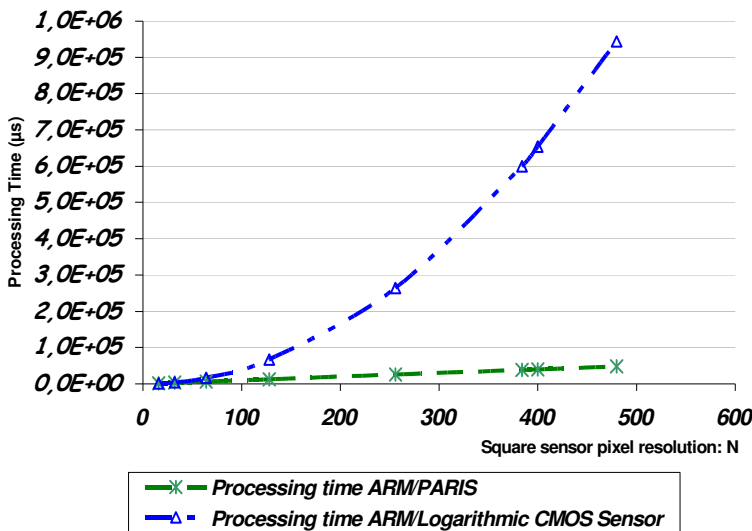


Fig. 19. Time processing of an edge detection: PARIS architecture versus ARM/Logarithmic CMOS sensor

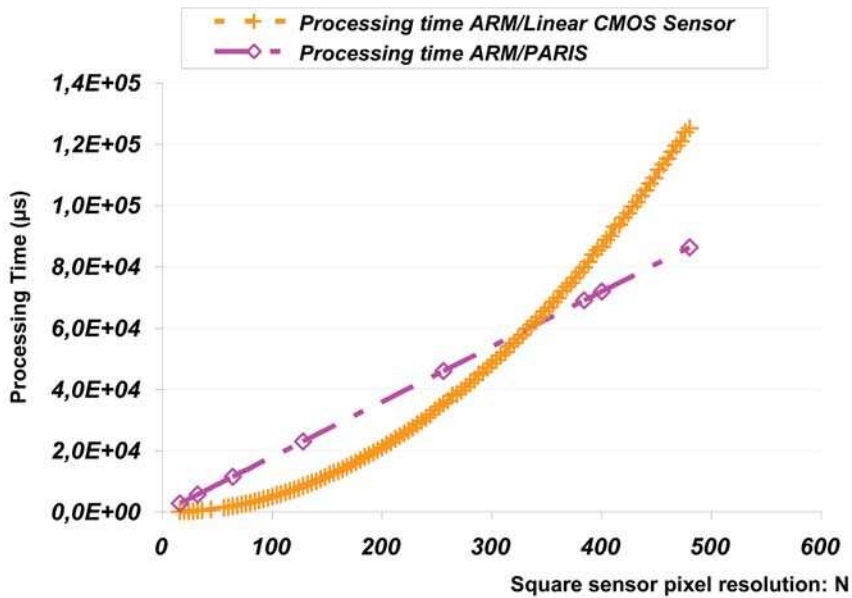


Fig. 20. Processing time of a Sobel operation: PARIS architecture versus ARM/Linear CMOS sensor

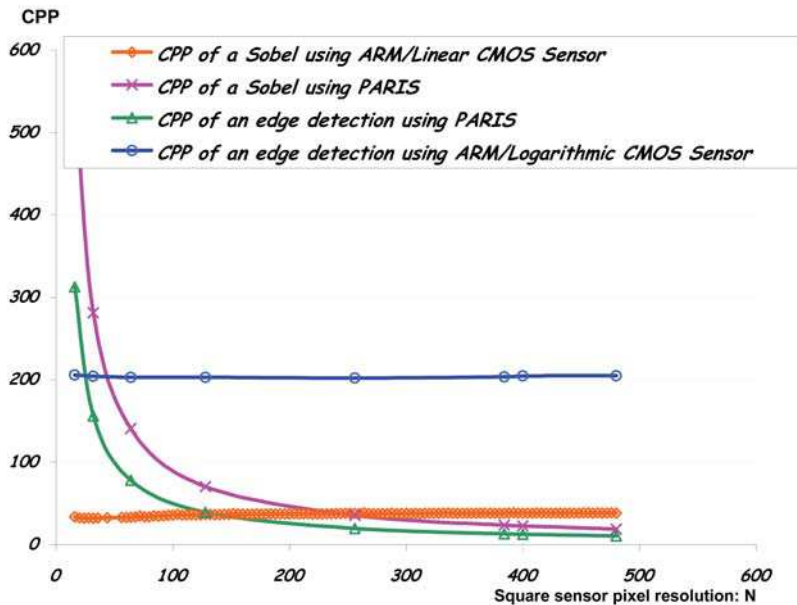


Fig. 21. Evolution of the CPP (Cycle Per Pixel) for PARIS and the ARM/CMOS architectures When 100 µW is the power consumption per 16 pixels and 300 µW is the power consumption per analogue processing unit. The 16x16 pixels circuit has a consumption of

50.4 mW. The consumption of the FUGA1000 sensor is 0.25 mW per pixel and that of the ARM microprocessor is 14 mW (RAM, ROM and logic glue consumption are excluded). It gives 76.5 mW consumption for 16x16 pixels resolution.

Hence, When comparing the power consumption between the FUGA1000/ARM like system and the PARIS retina at 10 MHz frequency, we conclude that the on chip solution allows better performances and low power consumption.

## 5. 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 chapter presents one experience of this concept named a retina.

It is concluded that on-chip image processing with retinas will offer benefits of low power consumption, fast processing frequency 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 retina based vision system. The goal is the integration of a microprocessor in the retina to manage the system and to optimise the hardware resources use.

To exhibit the feasibility of the chosen approach, we have presented an algorithm for the exposure time calibration. It is obvious that an algorithm of objects tracking, for example, will be more complex since the interval between two images is important.

As a result, if it is possible to carry out processed images in a short time, between two processing, the relevant objects will be seen as "immobile objects". Therefore, applications involving these algorithms will be less complex and efficient to implement them on a test bench. Our implementation demonstrates the advantages of the single chip solution and contributes as a highlight. Hence, designers and researchers can have a better understanding of smart sensing for intelligent vehicles (Elouardi, 2002, 2004). We propose implementing such a system with high resolution starting from a complex application on an intelligent vehicle embedding smart sensors for autonomous collision avoidance and objects tracking.

## 6. References

- Fossum R., "CMOS Image Sensors: Electronic Camera-On-A-Chip", IEEE Transactions on Electron Devices. Vol.44, No.10, pp.1689-98, Oct.97.
- Muramatsu, et al., "Image Processing Device for Automotive Vision Systems". Proceeding of IEEE Intelligent Vehicle Symposium 2002, Versailles, France.
- Burns R., Thomas C., Thomas P., Hornsey R., Pixel-parallel CMOS active pixel sensor for fast objects location, SPIE International Symposium on Optical Science and Technology, 3 - 8 Aug. 2003, San Diego, CA USA.
- El Gamal A., et al, "Pixel Level Processing: Why, what and how?" SPIE Vol.3650, 1999, pp. 2-13.
- Dudek P., Hicks J., "A CMOS General-Purpose Sampled-Data Analogue Microprocessor", Pro. of the 2000 IEEE International Symposium on Circuits and Systems. Geneva, Suisse.

- Ni Y., Guan J.H., "A 256x256-pixel Smart CMOS Image Sensor for Line based Stereo Vision Applications", IEEE, J. of Solid State Circuits, Vol. 35 No. 7, Juillet 2000, pp. 1055-1061.
- Alireza M., "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.
- Rowe A., Rosenberg C., Nourbakhsh I., "A Simple Low Cost Color Vision System," Technical Sketch Session of CVPR 2001, 2001.
- Seguine D., "Just add sensor - integrating analogue and digital signal conditioning in a programmable system on chip," Proceedings of IEEE Sensors, vol. 1, pp. 665-668, 2002.
- Koch C., Li H., "Vision Chips Implementing Vision Algorithms with Analogue VLSI circuits", IEEE Computer Society Press, 1995.
- Kleinfelder S., Lim S., "A 10 000 Frames/s CMOS Digital Pixel Sensor". IEEE Journal of Solid-State Circuits, Vol. 36, N° 12, Page 2049. December 2001.
- Arias-Estrada M., "A Real-time FPGA Architecture for Computer Vision", Journal of Electronic Imaging (SPIE - IS&T), Vol. 10, No. 1, January 2001, pp. 289-296.
- Zhou Z., Pain B., Fossum E., "Frame-transfer CMOS Active Pixel Sensor with pixel binning," IEEE Trans. On Electron Devices, Vol. ED-44, pp.1764-8, 1997.
- Dupret A., Klein J.O., Nshare A., "A DSP-like Analogue Processing Unit for Smart Image Sensors", International Journal of Circuit Theory and Applications 2002. 30: p. 595-609.
- Dupret A. et al, "A high current large bandwidth photosensor on standard CMOS Process" presented at EuroOpto'96, AFPAEC, Berlin, 1996.
- Dudek P., "A programmable focal-plane analogue processor array" Ph.D. thesis, University of Manchester Institute of Science and Technology (UMIST), May 2000.
- Dupret A., Klein J.O., Nshare A., "A programmable vision chip for CNN based algorithms". CNNA 2000, Catania, Italy: IEEE 00TH8509.
- Moutault S., et al, "Méthodologie d'analyse de performances pour les rétines artificielles". Report for Master graduation, IEF, Paris XI University. Orsay, 2000.
- Litwiller D., "CCD vs. CMOS: Facts and Fiction". The January 2001 issue of PHOTONICS SPECTRA, Laurin Publishing Co. Inc.
- Dierickx B., Bogaerts J., "Advanced developments in CMOS imaging", Fraunhofer IMS workshop, Duisburg, 25 May 2004.
- Walschap T. et al, "Brain Slice Imaging: a 100x100 Pixel CIS Combining 40k Frames per Second and 14 Bit Dynamic Range", IEEE Workshop on CCD & AIS, Elmau, 15-17 May 2003.
- Loose M., Meier K., Schemmel J., "CMOS image sensor with logarithmic response and self calibrating fixed pattern noise correction". Proc. SPIE 3410. ISBN 0-8194-2862-0, 1998, pp.117-127.
- Ogiers W. et al., "Compact CMOS Vision Systems for Space Use". [http://www.fillfactory.com/htm/technology/pdf/iris\\_cets.pdf](http://www.fillfactory.com/htm/technology/pdf/iris_cets.pdf)
- Bovik A., "Handbook of Image & Video Processing", Academic Press, 2000.
- Deriche R., "Fast algorithms for low level-vision". IEEE Transaction of Pattern Analysis and Machine Intelligence, vol 12-1, 1990.

- Garcia Lorca F., et al, "Efficient ASIC and FPGA implementation of IIR filters for real time edge detections", Proc. International Conference on Image Processing, IEEE ICIP 1997.
- Elouardi A., Bouaziz S., Reynaud R., "Evaluation of an artificial CMOS retina sensor for tracking systems". Pro. of IEEE IV'2002, Versailles, France.
- Elouardi A. et al, "Image Processing Vision System Implementing a Smart Sensor". Proceeding of IEEE Instrumentation and Measurement Technology Conference, IMTC'04. Pages 445-450. ISBN 0-7803-8249-8. May 18-20, 2004. Como, Italy.
- Langeheine J., et all "A CMOS FPTA Chip for Hardware Evolution of Analogue Electronic Circuits" Proceedings of the 2001 NASA/DoD Conference on Evolvable Hardware, pages 172-175, IEEE Computer Society, 2001.
- Schemmel J., Loose M., Meier K., "A 66 x 66 pixels analogue edge detection array with digital readout". Proceedings of the 25th European Solid-State Circuits 1999, Edition Frontinières, ISBN 2-86332-246-X, p. 298.
- Lew Yan Voona L.F.C., et all, "Real-Time Pattern Recognition Retina in CMOS Technology" Proceedings of the International Conference on Quality Control by Artificial Vision - QCAV'2001, Vol. 1, pp. 238-242, Le Creusot, FRANCE, May, 2001.
- Nudd G.R., et al., "A Charge-Coupled Device Image Processor for Smart Sensor Applications", SPIE Proc. Vol 155, pp.15-22, 1978.
- Knight T., "Design of an integrated optical sensor with on-chip processing", PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, Mass., 1983.
- Keast C.L., Sodini C.G., "A CCD/CMOS-based imager with integrated focal plane signal processing", IEEE Journal of Solid State Circuits, Vol. 28, No. 4, pp.431-7, 1993.



## **Image Processing**

Edited by Yung-Sheng Chen

ISBN 978-953-307-026-1

Hard cover, 516 pages

**Publisher** InTech

**Published online** 01, December, 2009

**Published in print edition** December, 2009

There are six sections in this book. The first section presents basic image processing techniques, such as image acquisition, storage, retrieval, transformation, filtering, and parallel computing. Then, some applications, such as road sign recognition, air quality monitoring, remote sensed image analysis, and diagnosis of industrial parts are considered. Subsequently, the application of image processing for the special eye examination and a newly three-dimensional digital camera are introduced. On the other hand, the section of medical imaging will show the applications of nuclear imaging, ultrasound imaging, and biology. The section of neural fuzzy presents the topics of image recognition, self-learning, image restoration, as well as evolutionary. The final section will show how to implement the hardware design based on the SoC or FPGA to accelerate image processing.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

A. Elouardi, S. Bouaziz, A. Dupret, L. Lacassagne, J.O. Klein and R. Reynaud (2009). Image Processing: towards a System on Chip, Image Processing, Yung-Sheng Chen (Ed.), ISBN: 978-953-307-026-1, InTech, Available from: <http://www.intechopen.com/books/image-processing/image-processing-towards-a-system-on-chip>

# **INTECH**

open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.