Gaussian Pyramid: Comparative Analysis of Hardware Architectures

Fernanda D. V. R. Oliveira, José Gabriel R. C. Gomes, Jorge Fernández-Berni,
Ricardo Carmona-Galán, Rocío del Río, Ángel Rodríguez-Vázquez
Universidade Federal do Rio de Janeiro – COPPE – PEE – Rio de Janeiro, RJ 21941-972
Instituto de Microelectrónica de Sevilla
E-mails: fernanda.dvro@poli.ufrj.br, gabriel@pads.ufrj.br, berni@imse-cnm.csic.es,
rcarmona@imse-cnm.csic.es, rocio@imse-cnm.csic.es, angel@imse-cnm.csic.es

***Abstract*—The paper addresses a comparison of architectures for hardware implementation of Gaussian image pyramids. Main differences between architectural choices are in the sensor front-end. One side is for architectures consisting of a conventional sensor that delivers digital images and which is followed by digital processors. The other side is for architectures employing a non-conventional sensor with per-pixel embedded pre-processing structures for Gaussian spatial filtering. This later choice belongs to the general category of "artificial retina" sensors which have been for long claimed as potentially advantageous for enhancing throughput and reducing energy consumption of vision systems. These advantages are very important in the internet of things context, where imaging systems are constantly exchanging information. The paper attempts to quantify these potential advantages within a design space in which the degrees of freedom are the number and type of ADCs (single-slope, SAR, cyclic, $\Sigma\Delta$ and pipeline), and the number of digital processors. Results show that speed and energy advantages of pre-processing sensors are not granted by default and are only realized through proper architectural design. The methodology presented for the comparison between focal-plane and digital approaches is a useful tool for imager design, allowing for the assessment of focal-plane processing advantages.**

I. Introduction

Images and *Vision*, i.e. the extraction of meaningful spatial-temporal information from the visual stimulus, are crucial for the interaction of "things" with the environment. Indeed, these days image and vision sensors are flooding all application territories, their usage and markets are increasing at exponential pace and they are expected to play important roles within Internet of Things domains [1], [2].

One major obstacle faced by image and vision sensor architects is the huge amount of *data* required for image coding. These data stress intermediate storage resources and communication channels. Also, their handling requires large energy budget particularly if *on-line* reaction is targeted. Different paths are being explored in the quest of overcoming these difficulties, covering from innovative sensor front-ends to enhanced *multi-core* back-end processors. Dynamic vision sensors [3] and computational image sensors [4]-[6] are relevant examples of advances regarding front-ends. In both cases sensors are meant to extract *information*, instead of just data, from the scene. Thus, reduced sets of *abstract* data, as opposed to raw data, are downloaded from the sensor for processing, hence de-stressing the system.

This paper deals with *information-centric* computational image sensors. Particularly a 6-T active pixel is proposed to expedite the calculation of the Gaussian Pyramid (GP) [7]. This is relevant because image pyramids, and in particular the GP, constitute the first stage of many computer vision processing pipelines [8]-[11]. Also, their computation mobilises significant computational resources and results into large delay and energy consumption. However, the functional primitive underlying GP calculation is rather simple — just *diffusions* across the scene plane are needed. Diffusions can be implemented by embedding simple mixed-signal circuits at pixel level, as done for instance in [12], [13]. Actually, results in [13] demonstrate orders of magnitude of improvement in throughput and energy consumption when compared to architectures using conventional, *data-centric*, sensors. But these advantages have the counterpart of much larger *pixel pitch*. The sensor hereby described aims at overcoming this drawback by using only two extra transistors per pixel; i.e. by employing a 6-T APS instead of the standard 4-T APS used in conventional image sensors [14].

The main asset of the 6-T GP pixel proposed in this paper comes from the parallel implementation of diffusions. However, data interchange requirements are still significant. It means that potential advantages of the non-conventional architecture are not granted by default. Exploring the conditions under which these advantages really occur, and benchmarking them, is the main purpose of this paper. In other words, we perform comparative throughput and energy analyses of a non-conventional architecture based on a 6-T GP pixel, on the one hand, and a conventional architecture, on the other hand. In this later architecture all processing for the GP takes place in the digital domain — no pre-processing at all is performed in the sensor. We consider different kinds of data readout and various architectural choices regarding the number of analog-to-digital converters (ADCs) embedded in the sensor readout channel, the type of ADC, and the number of processors in the digital back-end. Results presented in this paper show that potential advantages of the non-conventional architecture are largely dependent on the choice of these high-level, architectural degrees of freedom.

The paper is organized as follows: Sec. II defines the concept of Gaussian Pyramid and sets the context of its hardware realization; Sec. III addresses a pixel implementation where

only 6 transistors are required to perform GP processing; time and energy numerical comparisons are presented in Secs. IV and V, respectively; a case study is described in Sec. VI in order to validate the proposed implementation; finally some concluding remarks are presented in Sec. VII.

II. Gaussian Pyramid in Computer Vision

Object detection is the starting point for most computer vision pipelines. Once a particular object of interest is detected, it can be segmented, tracked, recognized etc. A major challenge for the implementation of this early vision task is that the scale of targeted objects is not known a priori. Objects can enter the surveyed scene at different distances from the image sensor. Those appearing at distant locations will require higher resolution to be detected than close-up objects for which most of the pixels will contain redundant information. The concept of *pyramid representation* [7] thus arises as a multi-resolution scene representation where each frame making up an image flow is progressively filtered and subsampled in order to efficiently deal with the search of objects at different scales. An example of pyramid is shown in Fig. 1(a). The images with no subsampling are depicted in Fig. 1(b) for better visualization of the applied filtering. Formally, filtering followed by subsampling is defined by the *reduce* operation given by:

$$f_l(i,j) = \sum_m \sum_n K(m,n) \cdot f_{l-1}(2i+m, 2j+n), \qquad (1)$$

where K is the filtering kernel and f_l is the image of the pyramid at level l [7]. The canonical way to construct a pyramid representation is based on Gaussian filtering [10]. This filter ensures that no artifacts are generated when going from finer to coarser scales. Indeed, *Gaussian Image Pyramid* is one of the predefined vision functions included in the industrial standard OpenVX [15]. We make use of this standard definition in our analysis.

Concerning hardware realization, a conventional approach to generate a GP is that of Fig. 2(a). The image sensed by an M×N pixel array is converted into digital and stored in memory. A prescribed number of Processing Elements (PEs) then access memory in order to process the image just captured and generate the corresponding pyramid. PEs can operate in parallel. This approach will constitute our reference realization for comparison.

Due to the significance of the GP as a fundamental processing primitive in computer vision, numerous non-conventional approaches aiming at boosting its hardware performance have also been reported [12], [13], [16]-[20]. Among them, mixed-signal focal-plane sensing-processing [12], [13], [16], [17] stands out as the best approach in terms of parallelization and energy efficiency. Additional circuitry is incorporated per pixel, usually connected to its counterpart at neighboring pixels, in order to concurrently process the image sensed by photo-sensitive circuit elements. Unfortunately, this approach typically suffers from large pixel pitch, thereby having a negative impact on key parameters of image sensing like sensitivity, resolution, noise etc.

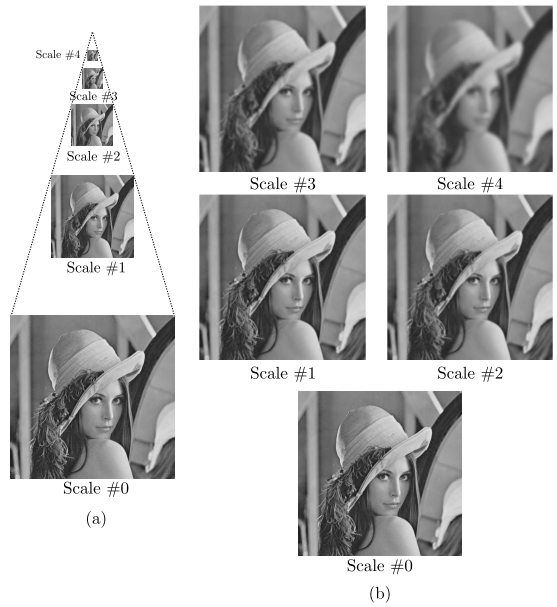

Fig. 1: (a) Example of pyramid representation; (b) Pyramid in (a) with no subsampling in order to highlight the effect of the applied filtering.

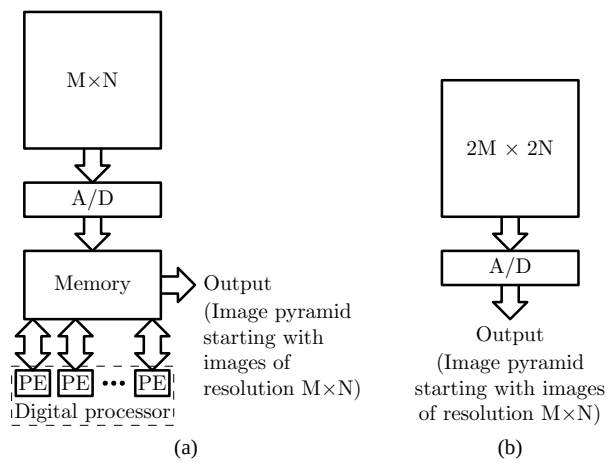

Fig. 2: Block diagram of two approaches for hardware realization of the Gaussian Pyramid: (a) conventional digital approach, where PE stands for Processing Element. PEs can operate in parallel; (b) focal-plane sensing-processing approach.

III. Proposed Pixel Implementation

In order to address this drawback of focal-plane sensing-processing realizations, we thoroughly analyze a focal-plane realization of Gaussian filtering requiring only two extra transistors per pixel. A basic block diagram of this realization is depicted in Fig. 2(b). The proposed circuit implementation is shown in Fig. 3. An n-channel transistor can be used as a switch, resulting in a pixel with six transistors [21]. Pixel operation starts by resetting the floating diffusion nodes. After the integration time, the charge accumulated at the photodiode cathode is transferred (according to TX) to the floating

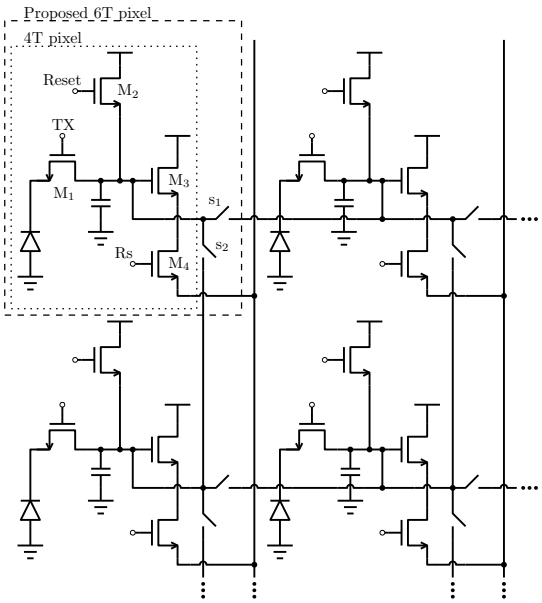

Fig. 3: 2×2 section of a matrix with the proposed pixel for focal-plane Gaussian filtering computation. Two transistors acting as switches (s_1 and s_2) are included inside each pixel to connect the floating diffusion nodes of neighboring pixels.

2	1	20	0	0	0	1	2
7	6	13	15	14	34	6	7
9	3	40	20	23	13	44	5
0	4	16	100	11	1	80	15
10	12	30	104	14	1	50	31
18	8	3	7	1	0	17	14
20	107	71	11	12	10	60	19
7	10	21	9	2	24	1	0

(a)

4	4	12	12	12	12	4	4
$p_{i-1,j-1}$		$p_{i-1,j}$		$p_{i-1,j+1}$		$p_{i-1,j+2}$	
4	4	12	12	12	12	4	4
4	4	44	44	12	12	36	36
$p_{i,j-1}$		$p_{i,j}$		$p_{i,j+1}$		$p_{i,j+2}$	
4	4	44	44	12	12	36	36
12	12	36	36	4	4	28	28
$p_{i+1,j-1}$		$p_{i+1,j}$		$p_{i+1,j+1}$		$p_{i+1,j+2}$	
12	12	36	36	4	4	28	28
36	36	28	28	12	12	20	20
$p_{i+2,j-1}$		$p_{i+2,j}$		$p_{i+2,j+1}$		$p_{i+2,j+2}$	
36	36	28	28	12	12	20	20

(b)

4	8	8	12	12	8	8	4
4	16	16	20	20	16	16	20
4	16	16	20	20	16	16	20
8	24	24	24	24	20	20	32
8	24	24	24	24	20	20	32
24	28	28	20	20	16	16	24
24	28	28	20	20	16	16	24
36	32	32	20	20	16	16	20

(c)

8	8	14	14	14	14	12	12
8	8	14	14	14	14	12	12
13	13	21	21	20	20	22	22
13	13	21	21	20	20	22	22
21	21	24	24	20	20	23	23
21	21	24	24	20	20	23	23
30	30	25	25	18	18	19	19
30	30	25	25	18	18	19	19

(d)

Fig. 4: Gaussian filtering example.

diffusion node. When the switches close, charge redistribution is performed among parasitic capacitors at the corresponding floating diffusion nodes. The average voltage after charge redistribution represents the mean luminance in the sub-matrix where the pixels were connected. This operation is lossy. Once pixels are interconnected in a sub-matrix, all parasitic capacitors end up holding the same voltage level. This loss of the original information does not prevent the GP generation.

As an example, consider the 8×8 matrix in Fig. 4(a), where the pixel values encode an original image. This first step consists in connecting the pixels into 2×2 blocks, to perform an average operation inside each block, as shown in Fig. 4(b). If we sample one pixel inside each block, then the resulting image has half the number of rows and half the number of columns of the original image. The first step is necessary to perform convolution in the proposed way, but it reduces the resolution of the image. Sub-sampled image pixel positions are written in $p_{i,j}$ format in the middle of each block in Fig. 4(b). All subsequent steps perform Gaussian filtering on this sub-sampled matrix. In the first subsequent step, we change the grid and, again, group the pixels into 2×2 blocks. This grid change and the result of the new charge redistribution step is shown in Fig. 4(c). After the charge redistribution we have that $p'_{i-1,j-1} = (p_{i-1,j-1} + p_{i-1,j} + p_{i,j-1} + p_{i,j})/4$, $p'_{i-1,j} = (p_{i-1,j} + p_{i-1,j+1} + p_{i,j} + p_{i,j+1})/4$, $p'_{i,j-1} = (p_{i,j-1}+p_{i,j}+p_{i+1,j-1}+p_{i+1,j})/4$ and $p'_{i,j} = (p_{i,j}+p_{i,j+1}+p_{i+1,j}+p_{i+1,j+1})/4$, where p' represents the pixel values after the second charge redistribution. The result is equivalent to filtering the sub-sampled image from Fig. 4(b) with the 2×2 binomial filter: $K = [1\ 1;\ 1\ 1]/4$.

If we change the grid again, back to the first grid, as shown in Fig. 4(d), we perform the same filtering for a second time. Changing the grid back to the first one corresponds to filtering the sub-sampled image from Fig. 4(c) with the 2×2 binomial, or equivalently, to filtering the sub-sampled image from Fig. 4(b) twice with the 2×2 binomial, or once with the 3×3 kernel: $K = [1\ 2\ 1;\ 2\ 4\ 2;\ 1\ 2\ 1]/16$.

According to the example in Fig. 4 we conclude that, for every grid change, the sub-sampled image is filtered with the 2×2 binomial kernel. The size of the targeted kernel determines the number of times that the grid must be shifted and charge redistribution enabled. The possible kernels that can be implemented with the proposed hardware are 2×2 binomial kernel cascade associations.

Figure 5 presents the steps required for the generation of a three-level pyramid according to the definition of GP of the standard OpenVX. Step (2) from Fig. 5, is required for changing the image resolution. To generate Level 0, which is the GP starting level, we sample one pixel inside each 2×2 block of the image generated after this charge redistribution. This image is then filtered through steps (4) to (7), resulting in the image that is subsampled to generate Level 1. To compute Level 2, we connect the pixels into 4×4 blocks, with the same goal of step (2), thus reducing the resolution. As in the calculation of Level 1, four charge redistribution operations are performed to filter the image, which is done in steps (10) to (13). By the end of these operations the result is subsampled, generating Level 2. To create a pyramid with four levels, the pixels are connected into 8×8 pixel blocks. The maximum number of levels that can be generated by the proposed hardware mainly depends on the fabrication technology leakage current and the floating diffusion node capacitance.

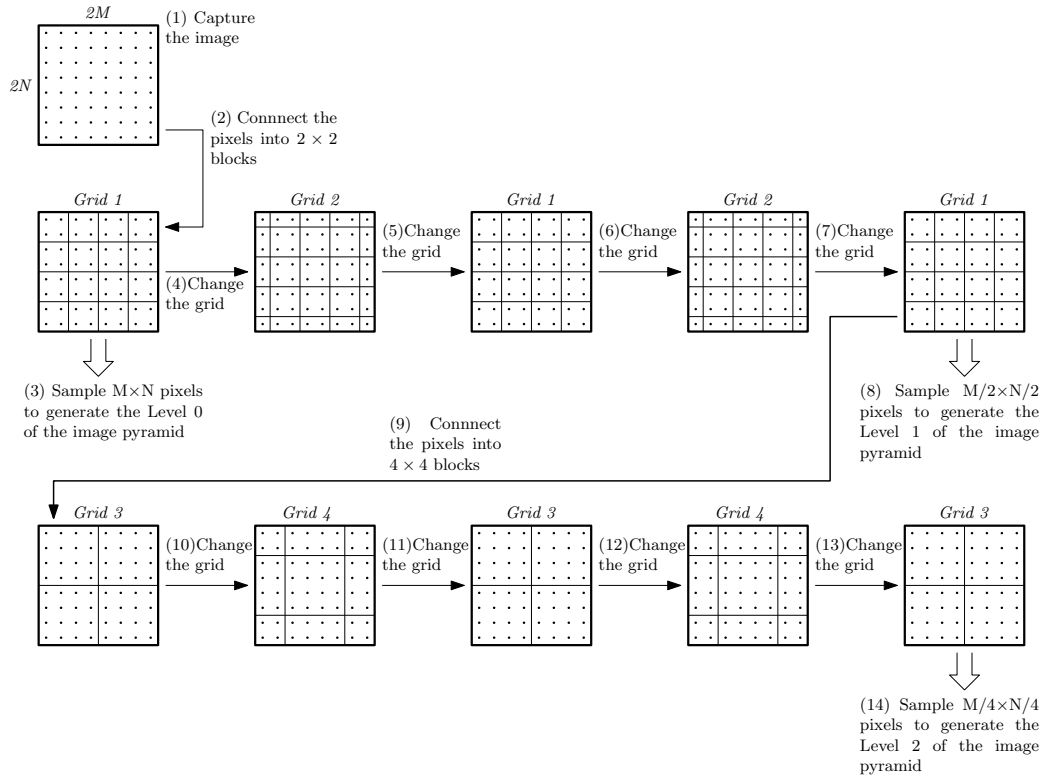

Fig. 5: Example of Gaussian Pyramid generation at the focal plane.

IV. Time Analysis Comparison

The main goal of this paper is to compare our reference digital implementation — depicted in Fig. 2(a) — to the focal-plane approach just described — sketched in Fig. 2(b). Note that for the focal-plane realization the resolution of the Level-0 filtered image (M×N) is a quarter of the resolution of the captured image (2M×2N).

In the digital processor, the convolution is based on sliding a binomial kernel across the image. At every location, the image pixels inside the kernel window are multiplied by the kernel elements, and the multiplication results are summed. For efficiency, the digital processor has a multiply and accumulate (MAC) unit, formed by one or more PE. The binomial kernel only requires addition and division by four, so the MAC unit is realized by simple digital circuitry (logic adders and shift registers) placed outside the pixel array. Filtering with a 2×2 kernel requires four pixel values for each kernel window, but two of these values are kept from the previous window operation, requiring only two memory-read accesses per window. Likewise, one MAC operation per window can be spared if we consider a partial result from the previous window. After each window computation the memory is accessed for writing the result.

For a numerical comparison, the flows of both architectures are broken into tasks, which are analyzed considering processing time and energy consumption. In the time analysis, each task is related to a variable τ that represents the time to perform a given task once. We then compute the number of times the task is executed. Overall time is equal to τ multiplied by the number of executions of that task. After finding the processing time expressions for both approaches as functions of τ, each τ is associated with the clock period, τ_{Clk}, which leads to expressions with a single global variable. The time needed for image capture is approximately the same for both approaches, so it is not considered in the time comparison. The same idea applies to the data output transmission.

A. Focal-Plane Approach Time Analysis

The focal-plane approach steps are inferred from Figs. 5 and 2(a). Aside from capture and transmission, there are two main steps:

1) Gaussian Pyramid generation: the time it takes to generate the GP depends on the number of charge redistribution operations multiplied by the time it takes for a single charge redistribution. Image size does not affect the GP generation time, because this operation runs concurrently across the matrix. Kernel size determines the number of charge redistributions per level. We need $n_k - 1$ charge redistribution operations to implement an $n_k \times n_k$ kernel. From Fig. 5 we see that this operation is repeated at every level, except the last one. Finally, we sum the charge redistribution operations that take place when the pyramid level changes. The overall number of charge redistribution operations is $N_{CR} = (N_{Lev} - 1) \cdot (n_k - 1) + (N_{Lev} - 1) = n_k (N_{Lev} - 1)$, where N_{Lev} is the number of pyramid levels. Multiplying N_{CR} by the time required for performing one charge redistribution, τ_{CR}, we have the overall processing time $\tau_{FP_{Proc}} = n_k (N_{Lev} - 1) \cdot \tau_{CR}$.

2) Analog-to-digital conversion: after each computation at the focal plane, pixel values are read out and sent to an analog-to-digital conversion stage, which comprises one or more ADCs. The time required for performing one sample conversion by one ADC is τ_{ADC}. Overall data conversion time depends on the number of ADCs, N_{ADC}, and on the amount of data converted, N_{conv}. To compute N_{conv}, we note that for every pyramid level the image size is reduced by a factor of 4: $N_{conv} = MN + MN/4 + \ldots + MN/2^{2(N_{Lev}-1)}$.
Overall conversion time is thus:

$$\tau_{ADC_{Total}} = \sum_{n=1}^{N_{Lev}} \frac{M \cdot N}{2^{2(n-1)}} \cdot \frac{\tau_{ADC}}{N_{ADC}}. \quad (2)$$

Overall focal-plane processing time is obtained by adding up $\tau_{FP_{Proc}}$ and $\tau_{ADC_{Total}}$:

$$\tau_{FP_{Total}} = n_k \left(N_{Lev} - 1\right) \tau_{CR} + \sum_{n=1}^{N_{Lev}} \frac{M \cdot N}{2^{2(n-1)}} \cdot \frac{\tau_{ADC}}{N_{ADC}} \quad (3)$$

B. Digital Implementation Time Analysis

The digital approach requires more steps than the focal-plane approach, as it can be seen in Fig. 2:

1) Analog-to-digital conversion: the captured image is immediately converted to digital. This is the only data conversion required by this approach. The size of the converted data is equal to the pixel array size. Thus, $\tau_{ADC_{Total}} = M \cdot N \cdot \tau_{ADC}/N_{ADC}$.
2) Memory storage: the resulting M×N digital values are stored into an internal memory. Time taken by this step is $M \cdot N \cdot \tau_{Mem}$, where τ_{Mem} is the time required for accessing a single memory position. To consider simultaneous memory access, we introduce a new variable, N_{busMem}, that represents the number of possible parallel accesses. The total time required by this step is $\tau_{matrixMemWrite} = M \cdot N \cdot \tau_{Mem}/N_{busMem}$.
3) Gaussian Pyramid generation: the digital processor reads input values for the current pyramid level from a memory, performs multiply and accumulate operations and writes the result back into the memory. The number of times this operation is performed depends on image size and on the number of times the image is filtered by the binomial kernel inside each level. Image size changes at every level according to a series similar to the one given for the number of conversions, N_{conv}, except for the fact that we do not perform convolutions at the highest level. The number of operations is equal to:

$$N_{op} = (n_k - 1) \cdot \sum_{n=1}^{N_{Lev-1}} \frac{M \cdot N}{2^{2(n-1)}}. \quad (4)$$

At least two pixel values are necessary in every computation of the 2×2 binomial kernel convolution, so we define $\tau_{memRead} = 2\tau_{Mem}$. At least three multiply and accumulate operations are used in the 2×2 kernel. The time required for performing these operations by one MAC unit is $\tau_{convolutionWindow} = 3\tau_{op}$, where τ_{op} is the time required by a single MAC operation. The resulting value is written in the memory through a single access, and so $\tau_{memWrite} = \tau_{Mem}$.
The time needed by a single PE to perform the convolution is obtained by multiplying the number of operations by the sum $(\tau_{memRead} + \tau_{convolutionWindow} + \tau_{memWrite})$. Assuming that more than one PE is available, and that N_{busMem} simultaneous memory accesses are allowed, parallel convolution operations are carried out. The overall time required for performing the convolution operations is, then, $\tau_{convolution} = N_{op}(2\tau_{Mem}/N_{busMem} + 3\tau_{op}/N_{PE} + \tau_{Mem}/N_{busMem})$. If $N_{PE} > N_{busMem}$, then memory access collisions occur. To simplify the analysis, we ignore this issue by assuming that every PE may access the memory at any moment, with no additional hardware complexity. Then, in the $\tau_{convolution}$ equation, N_{busMem} is substituted by N_{PE}: $\tau_{convolution} = N_{op}(2\tau_{Mem} + 3\tau_{op} + \tau_{Mem})/N_{PE}$.

By adding $\tau_{ADC_{Total}}$, $\tau_{MatrixMemWrite}$, and $\tau_{convolution}$, we have the digital approach overall time:

$$\tau_{digital_{Total}} = M \cdot N \cdot \frac{\tau_{ADC}}{N_{ADC}} + M \cdot N \cdot \frac{\tau_{Mem}}{N_{busMem}} + (n_k - 1) \cdot \sum_{n=1}^{N_{Lev-1}} \frac{M \cdot N}{2^{2(n-1)}} \left(\frac{2\tau_{Mem} + 3\tau_{op} + \tau_{Mem}}{N_{PE}} \right) \quad (5)$$

C. ADC Architectures Comparison

Before using the above equations to compare focal-plane and digital approaches, it is important to remember that it is common to work with the ADC at a clock period different from the one used for the other parts of the circuit. In our case, we define τ_{Clk} as the period of the clock signal that controls the pixel array, memory, and digital circuitry. The ADC clock period, on the other hand, is $K_{ADC} \cdot \tau_{Clk}$, where K_{ADC} depends on ADC type.

We consider five ADCs commonly used in CMOS image sensors: ramp, successive approximation register (SAR), sigma-delta (ΣΔ), cyclic and pipeline [22]. To compare ADC types and find the appropriate clock period in each case, we use reported imagers in which the performance figures of the embedded ADCs are included [23]-[58]. ADCs have already been compared by different authors [22], [59]. The present comparison focuses exclusively on ADCs designed for image sensors, in the context of comparative time and energy analysis, including recently published works.

The ramp converter, a linear approximation converter with simple architecture requiring low area and low power consumption [60], is probably the most used converter in image sensor applications [36]-[46]. It is suitable for working with high clock frequencies. We thus use it as a reference for other converter types: the ramp ADC clock period is equal to the global clock, $\tau_{Clk_{Ramp}} = K_{Ramp} \cdot \tau_{Clk} = \tau_{Clk}$, so

$K_{Ramp} = 1$. The data converters in the comparison were designed for different resolutions. For a fair comparison, we normalize the conversion rates and energies for the same number of bits, which is set as $N_{bits} = 8$. Although imagers with higher number of bits are common, eight bits per pixel is more typical [61]. The conversion rate normalization depends on the number of clock cycles per bit each converter architecture requires. A single slope ramp converter, for example, requires $2^{N_{bits}} \cdot \tau_{Clk_{Ramp}}$ (maximum) for a conversion. The normalized conversion rate considering eight bits is $f'_s = 2^{N_{bits}} \cdot f_s/2^8$, where f_s and N_{bits} are the reported conversion rate and resolution. For the SAR and cyclic converters, the conversion time is $N_{bits} \cdot \tau_{Clk_{SAR,Cyclic}}$, so the normalization is $f'_s = N_{bits} \cdot f_s/8$. The $\Sigma\Delta$ conversion time depends on the oversampling rate (OSR). For second-order incremental $\Sigma\Delta$ converters, the number of bits is $N_{bits} = \log_2[\text{OSR} \cdot (\text{OSR}+1)] - 1$, where OSR is the reported oversampling rate. We consider an oversampling rate equal to 25, which yields resolution equal to 8.3 bits. The normalization is $f'_s = \text{OSR} \cdot f_s/25$. The pipeline converter conversion time is one $\tau_{Clk_{Pipeline}}$, with some latency, which does not depend on the number of bits, i.e. normalization is not required. Pipeline converters are not as common in image sensors as the other converter types (simulation results have been reported, as well as experimental results from ADC chips working together with imaging chips), but they are included in the comparison because of their improved speed.

To normalize energy figures, we assume that the power consumption doubles for every bit added [59]: $E = 2^8 \cdot P/(f'_s \cdot 2^{N_{bits}})$. Walden's figure of merit for ADCs [62] uses the effective number of bits (ENOB) instead of the resolution. The normalized energy values in Fig. 6 are based on the resolution because some of the references do not report ENOB. Figure 6 shows the normalized energy versus normalized conversion rate for the five ADC types considered. The median conversion rate and energy (black markers in the figure) are chosen as representative values for each converter type. The median values suggest that, for eight-bit resolution, cyclic and SAR converters are approximately two times faster than ramp converters. The conversion times are related according to $\tau_{ADC_{Ramp}} = 2 \cdot \tau_{ADC_{SAR,Cyclic}}$ and $\tau_{ADC_{Ramp}} = 2^8 \cdot \tau_{Clk_{Ramp}}$, $\tau_{ADC_{SAR,Cyclic}} = 8 \cdot \tau_{Clk_{SAR,Cyclic}}$. So, the cyclic or SAR converters run at a clock which is approximately 16 times slower than the ramp converter clock. For the focal-plane and digital approaches comparison, we thus assume $K_{SAR,Cyclic} = 16$, where $K_{SAR,Cyclic}$ is the constant that multiplies the global clock period τ_{Clk} to yield $\tau_{Clk_{SAR,Cyclic}}$. The $\Sigma\Delta$ conversion time is 1.3 times smaller than the ramp ADC conversion time, so $\tau_{Clk_{\Sigma\Delta}} = 2^8 \cdot \tau_{Clk_{Ramp}}/(1.3 \cdot 25) \approx 8\tau_{Clk_{Ramp}}$. The multiplying constant is $K_{\Sigma\Delta} = 8$. For the pipeline converter, $\tau_{Clk_{Pipeline}} = [2^8/(\tau_{ADC_{Ramp}}/\tau_{ADC_{Pipeline}})]\tau_{Clk_{Ramp}}$ and $\tau_{ADC_{Ramp}}/\tau_{ADC_{Pipeline}} \approx 130$, $K_{Pipeline} = 2$.

Summarizing, we defined $K_{ramp} = 1$, since this converter is used as reference, and, using reported figures, found $K_{SAR,Cyclic} = 16$, $K_{\Sigma\Delta} = 8$ and $K_{Pipeline} = 2$. These constants define the ratio between the ADC clock period and the clock period τ_{Clk}, used for the other stages of the circuit.

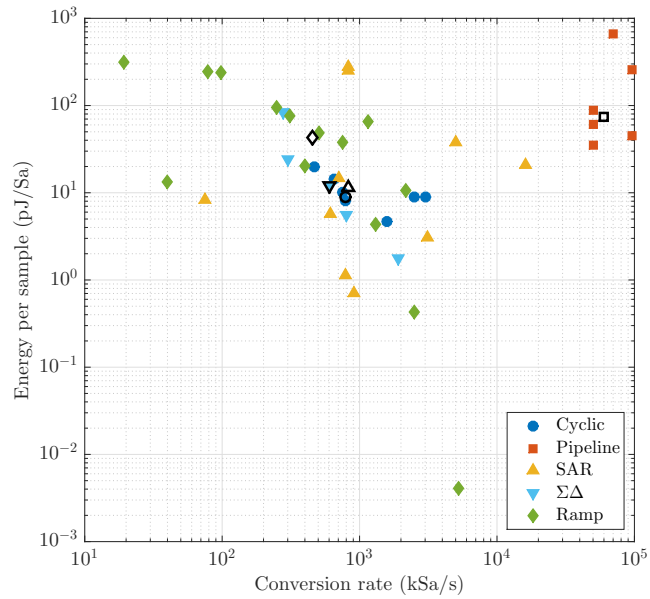

Fig. 6: Eight-bit normalized conversion rate versus energy per sample of five types of ADC. The median values for each type of ADC are plotted using black unfilled markers of the corresponding shape.

D. Time Comparison Results

We now establish some default values for the parameters in Eqs. (3) and (5), and associate the overall times to a global clock period. As explained in Sec. IV-C, τ_{Clk} is the period of the clock signal that controls the pixel array, memory, and digital circuitry and $K_{ADC} \cdot \tau_{Clk}$ is the ADC clock period.

Assuming that charge redistribution is practically instantaneous, it is clear from Eq. (3) that the bottleneck of the focal-plane approach is at the ADC, because of the amount of data to be converted. The digital approach bottleneck, on the other hand, is either at the ADC or at the processing stage, which depends on ADC type. For both approaches, we explore different ADC types and N_{ADC} values. For the digital approach, we explore several N_{PE}. We thus do not define default values for τ_{ADC}, N_{ADC}, and N_{PE}. The maximum N_{ADC} value is set to the number of columns at pyramid Level 0, since image sensors with one ADC per column are commonly found [63]. Although stacking technologies allow for the integration of one ADC per pixel [23], this is still an upcoming technology with high fabrication costs.

We use VGA (video graphics array, 640×480 pixels) standard for the pyramid Level 0 image size. Consequently, the pixel array size in the focal-plane approach is 1280×960. The time analysis does not change significantly if the resolution increases, but the bandwidth for the transmission of the generated data increases. Increasing the resolution and using one ADC per column also increases power consumption. The pyramid size can not be too large, because computation accuracy is limited by leakage currents. The operations can be performed as long as the capacitance voltages are not affected by these currents. We set $N_{Lev} = 4$. To achieve a

TABLE I: Time analysis equations parameters.

Parameter	Value
Pyramid Level 0 size (M×N)	640×480
Maximum number of ADCs ($N_{ADC_{Max}}$)	640
Equivalent kernel size (n_k)	5
Number of bits (N_{bits})	8
Number of levels (N_{Lev})	4
Number of memory accesses (N_{busMem})	4
Time to perform charge redistribution (τ_{CR})	$1\tau_{Clk}$
Time to access the memory (τ_{Mem})	$2\tau_{Clk}$
Time to perform a MAC operation (τ_{op})	$2\tau_{Clk}$

reasonable compromise between the circuit complexity and speed, we set $N_{busMem} = 4$. Choosing $N_{busMem} = 1$ would impair digital circuit performance, but increasing the number of simultaneous memory accesses increases digital circuit size and complexity.

Charge redistribution, memory access and MAC operation times (τ_{CR}, τ_{mem} and τ_{op}) are written as functions of the clock period τ_{Clk}. Charge redistribution itself is practically instantaneous, but the time it takes to drive the charge redistribution switches is considered, so $\tau_{CR} = 1\tau_{Clk}$. The time to access the memory, τ_{mem}, was defined as $2\tau_{Clk}$ considering that one clock period is necessary to define the position of the memory access and another to actually access that position. The time to perform a MAC operation, τ_{op}, was also defined as $2\tau_{Clk}$, since two clock cycles are necessary to perform the division by four operation and that the sum is performed with combinational logic, which does not depend on the clock. Table I summarizes the established parameter values.

Applying the parameter values in Eqs. (3) and (5) yields:

$$\tau_{FP} = 15\tau_{Clk} + \frac{640 \cdot 480 \cdot 1.33\tau_{ADC}}{N_{ADC_{FP}}}, \text{and} \tag{6}$$

$$\tau_{digital} = 640 \cdot 480 \left(\frac{\tau_{ADC}}{N_{ADC_{Dig}}} + \frac{\tau_{Clk}}{2} + \frac{63\tau_{Clk}}{N_{PE}} \right). \tag{7}$$

Equations (6) and (7) allow different N_{ADC} values for focal-plane and digital approaches. Charge redistribution time is not taken into account, because of its negligible contribution to Eq. (6). The ratio between the expressions in Eqs. (7) and (6) is:

$$\frac{\tau_{digital}}{\tau_{FP}} = \frac{\left(\frac{\tau_{ADC}/\tau_{Clk}}{N_{ADC_{Dig}}} + \frac{1}{2} + \frac{63}{N_{PE}} \right)}{\frac{1.33\tau_{ADC}/\tau_{Clk}}{N_{ADC_{FP}}}}. \tag{8}$$

Using the K_{ADC} constants defined in Sec. IV-C, we replace τ_{ADC} in Eq. (8) by an appropriate function of τ_{Clk}, which depends on the converter architecture. For the ramp converter we have $\tau_{ADC} = 2^8 \cdot K_{Ramp} \cdot \tau_{Clk} = 256\tau_{Clk}$. Considering that both the focal-plane and digital approaches use the ramp converter, the maximum advantage that the focal-plane approach achieves occurs when $N_{ADC_{Dig}} = 1$, $N_{PE} = 1$ and $N_{ADC_{FP}} = N_{ADC_{Max}} = 640$. The focal-plane approach is then 600 times faster than the digital approach. If $N_{ADC_{Dig}} = N_{ADC_{FP}} = N_{ADC_{Max}} = 640$, the focal-plane approach is 120 times faster. For ramp converters, the effect of increasing the number of PEs is shown in Fig. 7, in dash-dotted line, where the ratio between digital and focal-plane total operation times is plotted. With only four PEs, the focal-plane approach is 31 times faster, so for ramp ADCs the focal plane advantage is modest.

For the SAR or cyclic converters, we have $\tau_{ADC} = N_{bits} \cdot K_{SAR,Cylic} \cdot \tau_{Clk} = 128 \cdot \tau_{Clk}$. These converters require fewer clock cycles to perform one conversion, but their operation frequency is limited, hence resulting in performance comparable to that of the ramp ADC. The maximum advantage the focal plane achieves with SAR or cyclic converters corresponds to 700 times faster. The dashed line in Fig. 7 shows the evaluation of Eq. (8) for the SAR converter when $N_{ADC_{Dig}} = N_{ADC_{FP}} = N_{ADC_{Max}} = 640$. To reduce the advantage of the focal plane to less than two orders of magnitude, three PEs are necessary. With ten PEs, the focal-plane approach is 28 times faster. The $\Sigma\Delta$ conversion time depends on the OSR, which is equal to 25, as explained in Sec. IV-C: $\tau_{ADC} = \text{OSR} \cdot K_{\Sigma\Delta} \cdot \tau_{Clk} = 200 \cdot \tau_{Clk}$. The dotted line in Fig. 7 shows the comparison between focal-plane and digital approaches when the $\Sigma\Delta$ converter is used. The result is in between the ramp converter and the SAR converters: only two PEs are necessary to reduce the advantage of the focal plane to less than two orders of magnitude. For the pipeline converter analysis, we assume that it is not possible to integrate 640 converters inside the chip, because an imager with one pipeline converter per column has not been reported, to the best of our knowledge. For this converter, $\tau_{ADC} = K_{Pipeline} \cdot \tau_{Clk} = 2 \cdot \tau_{Clk}$. The solid lines in Fig. 7 correspond to results considering different numbers of pipeline ADCs. The focal-plane approach is highly advantageous when the number of ADCs is higher than 64. In this case, 18 PEs are necessary to drop the focal plane advantage to less than two orders of magnitude.

The speed of the digital processor may be increased by using double data rate (DDR), which allows for memory access and shift operation (division by four) to be carried out in a single clock period. In order to perform timing comparisons between the focal-plane approach and generic digital circuits not having additional power or area requirements, we do not take the DDR into account in the analysis. Nevertheless, if $\tau_{mem} = \tau_{op} = \tau_{Clk}$, the processing time ratios presented in Fig. 7 halve.

While focal-plane processing is being performed it is not possible to capture a new frame, which limits the frame rate. Even though, we can guarantee that the frame is always way above 30 frames/sec for the VGA resolution. If we consider a 100 MHz global clock, and one ramp converter per column, then approximately 1600 μs are necessary for generating the GP. Assuming that the image capture requires an additional 400 μs, then 2000 μs are necessary for image capture and GP generation, which yields frame rate around 500 fps. If the image resolution is increased to 6400×4800 (a factor of 100), it is still possible to achieve 60 fps by keeping the same conditions, which are namely one ramp converter per column and a global clock frequency of 100 MHz.

V. Energy Analysis Comparison

The energy analysis is more complicated because it is highly dependent on the architecture, the technology parameters are

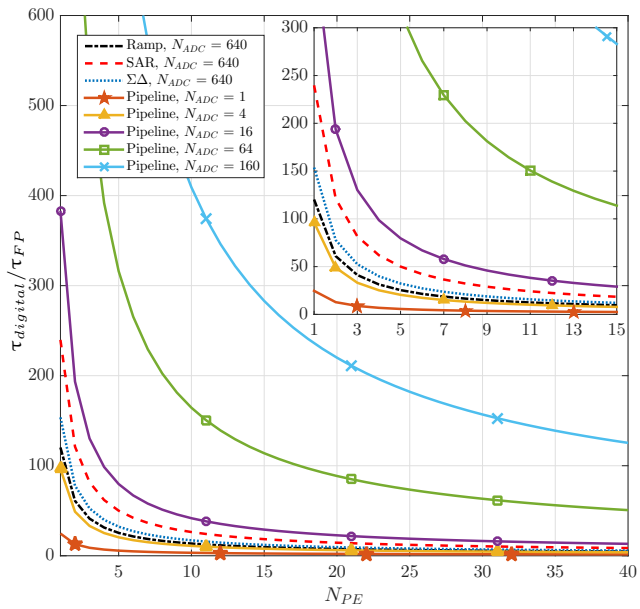

Fig. 7: Ratio between digital and focal-plane processing times as a function of the number of PEs. Ramp, SAR, ΣΔ, and pipeline ADCs are shown, respectively, in dash-dotted, dashed, dotted, and solid lines. For better visualization, a zoom of the curves is presented in the top right of the figure.

also of major importance and there is no global parameter (as the clock period was global in the time analysis). Also, aside from the stages necessary for the GP generation in each approach, both architectures must comprise the controlling circuits outside the pixel matrix, which are responsible for the interface between each stage shown in Fig. 2. Although these circuits play an important part on the energy consumption, a proper energy analysis of the controlling circuitry requires a careful design of this stage, which is not under the scope of this paper, so these circuits are not considered.

For the ADC stage, the energy consumption depends on the type of converter and architecture. A general empirical analysis on the energy efficiency of ADC architectures can be found in [64]. This paper defines a lower boundary for energy consumption per sample equal to $2^{2(ENOB-9)}$, and states that lowering the resolution below nine bits results in minor advantages. The minimum energy per sample in our case, eight bits, would be thus equal to 1 pJ/Sa. Although it is important to have this lower boundary limit it is also interesting to consider converters that have been used for image sensors. As mentioned in Sec. IV-C, several references were used for finding representative values of conversion rate and energy consumption for each ADC architecture. The median energy consumption per sample for each type of ADC, which can be seen in Fig. 6, is used in this section.

Aside from the ADC, the other sources of energy consumption can be divided in: DC consumption, E_{DC}, when there is a constant current flowing, usually for biasing circuits; dynamic consumption, $E_{Dynamic}$, as a result of the circuit activity, which requires charging and discharging capacitive nodes of the circuit; static consumption, E_{Static}, which is the energy

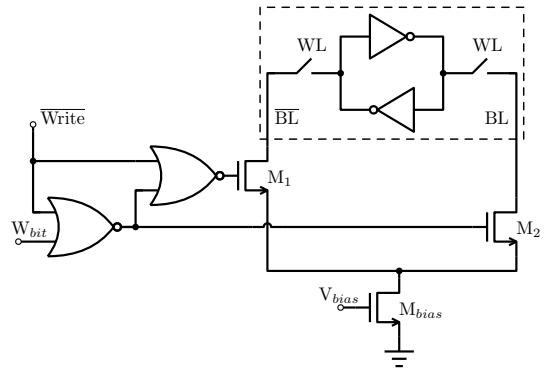

Fig. 8: One-bit SRAM memory cell, inside the dashed box, and memory write control circuit.

that the transistor consumes even when it is off, depending on the leakage current I_{leak}; and short-circuit consumption, $E_{Shortcircuit}$, which is another source of dynamic energy and happens when switching the inputs of a logic gate, in a moment when both n-channel and p-channel transistors are on, thus allowing for a short-circuit current to flow. The short-circuit current can be minimized by matching the rise/fall times of the input and output signals, reaching a maximum of 15% of the total dynamic consumption [65]. $E_{Shortcircuit}$ is computed as a portion of the dynamic energy: $E_{Shortcircuit} = 15(E_{Dynamic} + E_{Shortcircuit})/100 \rightarrow E_{Shortcircuit} = 15E_{Dynamic}/85$. In the following equations, C_n is the node capacitance, V_{ddM} is the pixel matrix voltage supply and V_{dd} is the voltage supply outside the pixel matrix.

The dynamic power consumed by a digital circuit can be estimated by $P_{dynamic} = N_d \cdot C_n \cdot V_{dd}^2 \cdot f_{0\rightarrow1}$ where N_d is the number of nodes and $f_{0\rightarrow1}$ is the switching frequency of the nodes from 0 to 1 [65]. This equation is found considering that every node in the digital circuit is capacitive and that the energy necessary to charge a capacitive node is equal to $C_n \cdot V_{dd}^2$. The switching frequency can be written as a function of the clock frequency: $f_{0\rightarrow1} = \alpha f_{clk} = \alpha/\tau_{Clk}$, where α is called switching activity factor and represents the probability of a node switching from 0 to 1, resulting in $P_{dynamic} = \alpha \cdot N_d \cdot C_n \cdot V_{dd}^2/\tau_{Clk}$. The energy is given by $P_{dynamic}$ multiplied by the time during which the circuit operates: $E_{dynamic} = \alpha \cdot N_d \cdot C_n \cdot V_{dd}^2 \tau_{total}/\tau_{Clk}$. In our case, τ_{total} can be computed according to the time analysis presented in Sec. IV.

The SRAM memory is considered for the energy analysis of the digital circuit. The schematic diagram of a one-bit cell of this memory is shown in Fig. 8. The memory has the same size of the Level 0 image in the pyramid, M×N, and each pixel is represented with N_{bits}. In order to read a value from the memory, we need to select the memory row using the switch WL and read the result in the BL bus. Writing requires selecting a memory cell through the WL switches and setting $\overline{\text{Write}}$ to zero, which closes transistor M_1 or M_2, depending on the bit that is being written, W_{bit}. If W_{bit} is logical zero, transistor M_2 closes and the bias current generated by V_{bias} discharges the bitline BL. If W_{bit} is logical one, transistor M_1 closes and the bias current discharges the bitline $\overline{\text{BL}}$ and thus charges BL.

A. Focal plane

Except for the A/D conversion stage, which was explained in the beginning of the section, the steps that were considered for the energy consumption estimation are described next. As opposed to the time analysis computation, here we have to consider the image capture and readout steps because the pixel matrix size has an influence in the consumption.

1) Image capture: this operation involves, for each pixel, charging the floating diffusion node and operating the Reset and TX switches, shown in Fig. 3. *Dynamic:* the energy for capturing a single pixel can be estimated as the one necessary for charging three capacitances, $E_{pixCapture} = (C_{FD} \cdot V_{ddM}^2) + (C_{Rst} \cdot V_{ddM}^2) + (C_{TX} \cdot V_{ddM}^2)$. Since this operation happens for every pixel of the matrix, $E_{capture} = 2M \cdot 2N \cdot E_{pixCapture}$. The capacitances C_{FD}, C_{Rst} and C_{TX} can be replaced by the node capacitance C_n, thus $E_{capture} = 2M \cdot 2N \cdot (3 \cdot C_n \cdot V_{ddM}^2)$. *Static:* transistors M_1 and M_2 from Fig. 3 are off for most of the operation and contribute with static energy consumption, $E_{matrixStatic} = 2(2M \cdot 2N \cdot V_{ddM} \cdot I_{leak} \cdot \tau_{FP_{Total}})$, where $\tau_{FP_{Total}}$ is given by Eq. (3).
2) Charge redistribution: this operation is passive, but energy is necessary to close the switches that connect the floating diffusion nodes. *Dynamic:* the energy that is needed to control two switches per pixel, $2C_n \cdot V_{ddM}^2$, must be multiplied by the number of times the charge redistribution is performed (from Sec. IV-A) and by the size of the pixel matrix, since the operation is performed throughout the entire matrix, $E_{CR} = (N_{Lev} - 1) n_k \cdot 2M \cdot 2N \cdot (2C_n \cdot V_{ddM}^2)$, where n_k is the size of the filter.
3) Image readout: reading a pixel requires closing the row select switch and enabling the current source that biases the source follower. This current flows for the time necessary to charge the pixel matrix column capacitance. *Dynamic:* the gate of transistor M_4, from Fig. 3, is connected to a bus with every other select transistor of the same row of the matrix, the equivalent capacitance is estimated as $2M \cdot C_n$. The pixel matrix column capacitance, on the other hand, depends on the number of rows and is estimated as $2N \cdot C_n$. The dynamic energy is thus $E_{pixelReadDynamic} = (2M + 2N) \cdot C_n \cdot V_{ddM}^2$. The pixel matrix columns capacitances are charged whenever a pixel is read. The number of times a pixel is read is equal to N_{conv}, defined in Sec. IV-A. The row select switch is activated every time the image is being read, once for each row, thus N_{conv}/M times. The total energy is $E_{readTotal} = [(N_{conv}/M) \cdot 2M + N_{conv} \cdot 2N] \cdot C_n \cdot V_{ddM}^2$.

B. Digital

For the digital approach, we have the following steps:

1) Image capture: following the same analysis as in the focal-plane case, but changing the image size, yields $E_{capture} = M \cdot N \cdot (3 \cdot C_n \cdot V_{ddM}^2)$ and $E_{matrixStatic} = 2(M \cdot N \cdot V_{ddM} \cdot I_{leak} \cdot \tau_{Digital})$.
2) Image readout: also very similar to the focal plane, but the bus capacitance changes and the image is read only once, $E_{readTotal} = (N \cdot M + M \cdot N \cdot N) \cdot C_n \cdot V_{ddM}^2$.
3) MAC operation: the digital processor that is considered is a MAC unit formed by a logic adder and a shift register. *Dynamic:* the energy consumed by a digital circuit was explained in the beginning of this section. In the case of the MAC operation, the time during which the circuit operates is $N_{op} \cdot 3\tau_{op}$ (according to Sec. IV-B), so $E_{MACdynamic} = \alpha \cdot N_d \cdot C_n \cdot V_{dd}^2 (N_{op} \cdot 3\tau_{op})/\tau_{Clk}$. *Static:* depends on the overall number of transistors inside the digital ports. Half of the transistors inside a common logic gate are off, so $E_{MACstatic} = N_{Off} \cdot V_{dd} \cdot I_{leak} \cdot \tau_{Digital}$. *Short-circuit:* as explained in the beginning of the section, $E_{MACshort} = 15 E_{MACdynamic}/85$.
4) Memory read: reading requires charging the WL bus capacitance C_{WL}, two switches per bit, and the BL or $\overline{\text{BL}}$ bus capacitance, represented by C_{BL}. *Dynamic:* $E_{readDyn} = (\alpha \cdot C_{BL} + C_{WL}) \cdot V_{dd}^2 \cdot N_{op} \cdot 2\tau_{mem}/\tau_{Clk}$, where $N_{op} \cdot 2$ is the number of times the memory is accessed for reading, according to Sec. IV-B. The activity factor α is only necessary for the BL bus and represents the cases where the bus voltage does not change when closing WL. The WL switch remains closed while the reading is performed and opens right after, so there is no activity factor in this case. *Static:* from Fig. 8, inside a one-bit memory cell, each inverter has one n-channel transistor and one p-channel transistor. Regardless of the state of the memory there is one p-channel transistor off and one n-channel transistor off. Besides, the WL switches can be formed by one n-channel transistor each, which are off most of the time. Thus, $E_{readStatic} = 4 \cdot V_{dd} \cdot I_{leak} \cdot \tau_{Digital}$. *Short-circuit:* $E_{readShortcircuit} = 15(E_{readDyn})/85$.
5) Memory write: writing a single value in the memory requires more energy than reading a single position of the memory because the bias current is activated, and the writing controlling circuits are used. *Dynamic:* $E_{writeDyn} = (\alpha \cdot C_{BL} + C_{WL} + \alpha \cdot C_{W_{bit}} + C_{\overline{\text{Write}}} + C_n) \cdot V_{dd}^2 \cdot N_{MemWrite} \cdot \tau_{mem}/\tau_{Clk}$, where $C_{W_{bit}}$ is the capacitance of the input W_{bit} of the controlling circuit, $C_{\overline{\text{Write}}}$ is the capacitance of the node $\overline{\text{Write}}$ and C_n is the gate capacitance of either M_1 or M_2, which are complementary nodes, so only one capacitance is considered. The number of times the memory is accessed for writing is $N_{MemWrite} = M \cdot N + N_{op}$, from Sec. IV-B. *Static:* the static power consumption is only due to the contribution of the write control circuit, because the cell circuit contribution was provided in item (4). Transistors M_1 and M_2 are on only when a bit is written, so we assume that they contribute with the static consumption during the entire operation. These transistors are necessary for every column of the memory matrix, so it must be multiplied by $N_{bits} \cdot M$. Furthermore, inside the NOR gates there is always two transistors off. We can consider that this circuit is repeated for each bit and for, at least, each N_{busMem}, resulting in $E_{writeStatic} = (M \cdot 2 \cdot C_n + N_{busMem} \cdot 4 \cdot C_n) \cdot N_{bits} \cdot V_{dd} \cdot I_{leak} \cdot \tau_{Digital}$. *Short-circuit:* $E_{writeShortcircuit} = 15(E_{writeDyn})/85$. *DC:* the bias current, that is activated whenever we need to swap a bit in the desired writing position, flows only

TABLE II: Energy analysis equations parameters.

Parameter	Value
Node capacitance (C_n)	4 fF
Matrix voltage supply (V_{dd})	3.3 V
Voltage supply outside the matrix (V_{ddM})	1.5 V
Leakage current (I_{leak})	2.6 pA
Memory $I_{biasMem}$	50 μA
Clock frequency	100 MHz
Activity factor (α)	0.2; 0.8
Ramp ADC energy	43 pJ/sample
$\Sigma\Delta$ ADC energy	12 pJ/sample
SAR ADC energy	11 pJ/sample
Cyclic ADC energy	9 pJ/sample
Pipeline ADC energy	74 pJ/sample

for the time necessary to discharge the bus capacitance, $E_{MemDC} = \alpha \cdot N_{memWrite} \cdot V_{dd} \cdot I_{biasMem} \cdot \tau_{Clk}/10$, where τ_{Clk} is divided by ten to model capacitance discharge time, which is significantly shorter than the clock period. The activity factor is necessary to represent the cases where the cell bit that is being written does not change.

C. Energy comparison

To compare focal-plane and digital approaches, we use the values shown in Tab. II. Node capacitance, voltage supply, leakage and memory bias current were established by means of simulations with a 110 nm CMOS technology. The clock frequency determines static energy consumption: 100 MHz is arbitrarily chosen, considering the clock frequency reported in some papers. The activity factor is $0 < \alpha \leq 1$ [65]. Two values were chosen for α to give an idea of how the energy changes according to it. An activity factor closer to one benefits the focal-plane approach. The energy of the converters are the median energy consumption values from Fig. 6.

Aside from the values defined in the table, it is also necessary to estimate the number of nodes of the MAC unit circuit. An example of a two-bit adder with carry and an eight-bit shift register is shown in Fig. 9. From the figures, we deduce that an N_{bits} adder requires at least $4 + 7 \cdot (N_{bits} - 1)$ nodes and the N_{bits} shift register at least N_{bits} nodes. Thus, a single PE of our MAC unit can be implemented with $(8 \cdot N_{bits} - 3)$ nodes. The flip-flop from Fig. 9 actually requires more nodes, but we are assuming N_{bits} nodes as an optimistic estimation, which benefits the digital approach.

Determining the memory node capacitances is also necessary for the comparison. The capacitance of the node $\overline{\text{Write}}$, $C_{\overline{\text{Write}}}$, is equal to $2C_n$, since $\overline{\text{Write}}$ is connected to two logic gate inputs. For the bit capacitance, considering that it is connected to a column bus, $C_{W_{bit}} = N \cdot C_n$. The bitline capacitance also depends on the number of rows, $C_{BL} = N \cdot C_n$. The wordline capacitance depends on the number of the memory matrix columns: $2 \cdot N_{bits} \cdot M \cdot C_n$.

Considering the values from Tab. II, $\alpha = 0.2$ and 640 converters for both approaches, the focal-plane approach requires 33 times less energy than the digital approach when the ramp converter is being used. For the SAR, cyclic and $\Sigma\Delta$ converters, the focal plane is around 52 times more energy-efficient. For the pipeline converter, the focal-plane approach is 24 times more energy-efficient. Making $\alpha = 0.8$, there is a modest increase in the advantage of the focal plane: it is 34, 54 and 25 times more energy efficient for the ramp, SAR (also cyclic and $\Sigma\Delta$) and pipeline, respectively.

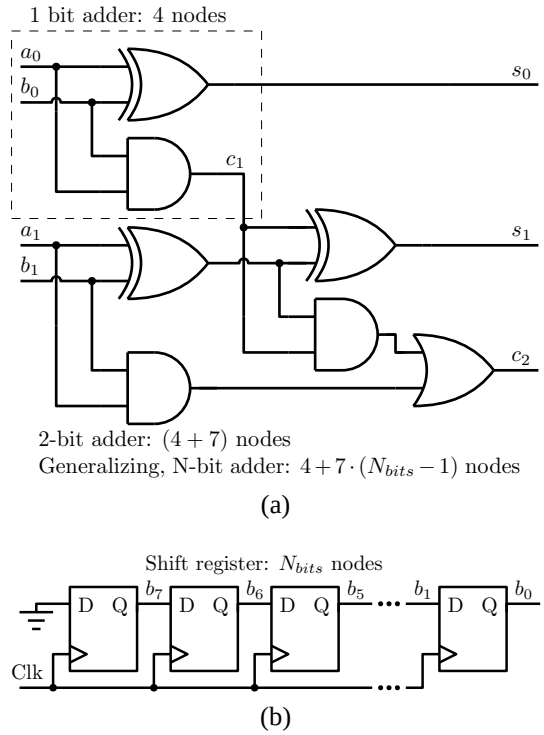

Fig. 9: Circuits considered for the MAC energy estimation: (a) N_{Bits} adder and (b) shift register.

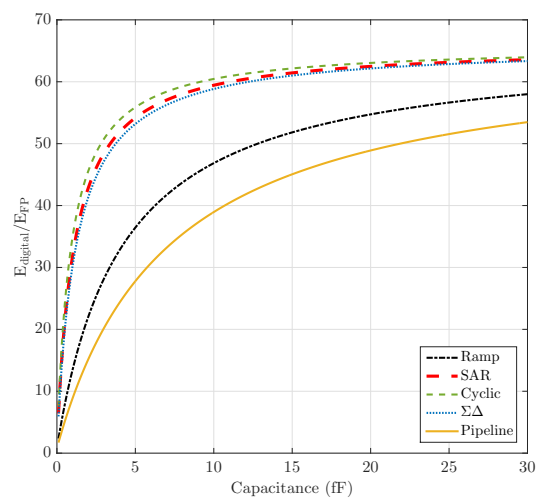

Fig. 10: Node capacitance effect on the energy consumption.

It is interesting to see the effect of the capacitance increase on the result. Since most of the nodes considered for the analysis are connected to metal input or output lines, the metal parasitic effects would probably result in capacitances higher than the ones considered. Figure 10 shows how the ratio between digital energy consumption and focal-plane energy consumption varies as the C_n of the nodes connected to metal lines increases. The activity factor used in this plot is 0.2.

Let us consider, for example, that we use the ADC presented in [30]. This is a column parallel SAR ADC that, normalized to eight bits, consumes 14.6 pJ per sample, with an ADC clock frequency of $\tau_{Clk_{SAR}} = 5.6$ MHz. Under these conditions, the focal-plane approach takes 911 μs to generate the GP. If we use 10 PEs in the digital approach, then the focal plane is 26 times faster. The energy consumed with the focal-plane approach is around to 23 μJ, 49 times more energy-efficient than the digital approach.

VI. Case Study: SIFT algorithm

The first step of the scale invariant feature transform (SIFT), which is an object recognition algorithm, is multiple-scale image representation [66]. First, the image is filtered n times with Gaussian kernels, thus creating the first octave. The image from the middle of the octave is then copied and subsampled. The resulting image is filtered with the same kernels of the first octave, thus generating the second octave. The procedure is repeated until the target number of octaves is obtained. A difference of Gaussian (DoG) is performed afterwards in order to create a scale-normalized Laplacian of Gaussian ($\sigma^2\Delta^2 G$) representation of the image. Points of interest are then searched throughout the scales of the Laplacian scale-space pyramid representation.

With the proposed hardware, it is possible to generate a scale space that can be used by the SIFT without a significant performance drop [21]. First, we capture the image and group the pixels into 2×2 pixel blocks. After sampling and quantization, the result is the first image from the first octave of the scale space. We then change the grid and obtain the second scale-space image. This kernel is a good approximation of the Gaussian kernel with standard deviation $\sigma_{filter} = \sigma_1 = 0.5$. By changing the grid again, we perform a second filtering operation, which results in the third image from the scale space. The resulting standard deviation is $\sigma_2 = \sqrt{\sigma_1^2 + \sigma_{filter}^2} = 0.707$. The ratio of the standard deviations of adjacent scale-space filters must be kept constant [21], $k = \sigma_2/\sigma_1 = \sqrt{2}$. Consequently, the next image must be the result of filtering with a kernel with standard deviation equal to $k \cdot \sigma_2 = 1$. This is achieved by using the binomial kernel twice: $\sqrt{\sigma_2^2 + \sigma_{filter}^2 + \sigma_{filter}^2} = 1$, which leads to the fourth image from the scale space. The next octave is computed after all the images from the previous octave are generated, by grouping the pixels into 4×4 blocks and repeating the filtering procedure.

System-level simulations show that the results achieved with the proposed hardware implementation are similar to those obtained with the original approach. These simulations were run using the database from [67] and OpenCV SIFT libraries. By computing original image keypoints and comparing them with transformed image keypoints, we evaluate whether the proposed keypoint method is robust to those transformations. This evaluation measure is denoted as *repeatability*.

Table III shows repeatability results for the original, fully digital, and the proposed, focal-plane, method. The original method parameters are: three octaves, six scales per octave, 0.04 for contrast threshold (which is used for removing weak features), and 10 for edge threshold (which is used for filtering edge-like features). For the focal-plane method, we also have three octaves, but four scales, 0.05 for contrast threshold (more selective), and the same edge threshold. As it can be seen in Tab. III, the systems yield similar results, which validates focal-plane hardware scale-space implementation for SIFT.

TABLE III: System level repeatability results.

Image	Bark		Bikes	
Transformation	Original	Proposed	Original	Proposed
H1to2	67.54%	65.77%	56.55%	76.57%
H1to3	62.76%	30.86%	57.06%	76.33%
H1to4	75.21%	23.70%	53.83%	73.40%
H1to5	73.09%	0.00%	55.29%	71.98%
H1to6	70.21%	9.82%	48.53%	67.78%
Image	Boat		Graf	
Transformation	Original	Proposed	Original	Proposed
H1to2	59.39%	69.11%	60.47%	55.76%
H1to3	60.06%	10.04%	48.15%	22.19%
H1to4	43.47%	36.59%	22.96%	8.37%
H1to5	41.26%	57.42%	0.00%	0.00%
H1to6	31.97%	5.87%	0.00%	0.00%
Image	Leuven		Trees	
Transformation	Original	Proposed	Original	Proposed
H1to2	63.99%	74.13%	51.47%	65.46%
H1to3	60.86%	75.34%	51.51%	64.54%
H1to4	60.34%	73.38%	44.17%	62.08%
H1to5	57.85%	71.92%	42.07%	66.28%
H1to6	52.49%	73.75%	38.49%	68.72%
Image	UBC		Wall	
Transformation	Original	Proposed	Original	Proposed
H1to2	67.86%	83.26%	61.82%	67.77%
H1to3	63.84%	77.46%	57.12%	62.57%
H1to4	57.54%	72.37%	52.95%	47.56%
H1to5	42.46%	64.84%	41.35%	34.91%
H1to6	40.44%	59.21%	10.10%	15.53%
Average repeatability: Original = 50.16%; Proposed = 51.07%				

The same time and energy analysis carried out in Secs. IV and V can be extended for scale-space generation. In this case, the image does not change resolution after each filtering operation (more convolutions are performed at the focal plane) and some specific images must be sampled. The conclusions remain the same: the scenario in which the focal-plane approach shows most advantage is the one in which fast converters are being used, when we have one data converter per column. The time equations obtained from the scale-space analysis using the ideas presented in Sec. IV are:

$$\tau_{FP_{Total}} = N_{oct} \cdot 2^{N_{scales}-2} \cdot \tau_{CR} + N_{oct} \cdot \tau_{CR} + \\ + \sum_{n=1}^{N_{oct}} \frac{(M \cdot N \cdot N_{scales})}{2^{2(n-1)}} \cdot \frac{\tau_{ADC}}{N_{ADC}}, \quad (9)$$

$$\tau_{digital_{Total}} = M \cdot N \cdot \frac{\tau_{ADC}}{N_{ADC}} + M \cdot N \cdot \frac{\tau_{Mem}}{N_{busMem}} + \\ \sum_{n=1}^{N_{oct}-1} \frac{M \cdot N}{2^{2(n-1)}} \cdot \left(\frac{4\tau_{Mem} + 4\tau_{op} + \tau_{Mem}}{N_{PE}} \right) + \\ 2^{N_{scales}-2} \sum_{n=1}^{N_{oct}} \frac{M \cdot N}{2^{2(n-1)}} \cdot \left(\frac{2\tau_{Mem} + 3\tau_{op} + \tau_{Mem}}{N_{PE}} \right), \quad (10)$$

where the number of scales is N_{scales} (greater than or equal to 2), and the number of octaves is N_{oct}. Within each octave, the number of charge redistribution operations is $2^{N_{scales}-2}$.

VII. Conclusions

Sensors with embedded per-pixel processors have been since long advocated as critical for increasing speed and decreasing energy consumption of vision hardware. These claims rely on two conceptual pillars: on the one hand, analog processing is known to have larger energy efficiency than digital for applications with moderate SNR requirements; on the other hand, sensor pre-processing features data compression at the sensor, thus relaxing bandwidth and storage requirements. The analyses that were carried out in this paper show that these potential advantages are case-specific. These analyses are completed for a vision primitive which is commonly employed in computer vision, namely the image pyramid. The computation of GPs can be accelerated by employing a non-conventional sensor front-end with extra per-pixel circuitry to perform spatial filtering. When comparing this approach with the use of a conventional sensor, without embedded pre-processing, followed by a conventional processor, a bottleneck of the former is found at the required number of analog-to-digital conversions. Different image sensors ADCs are considered in the paper with the goal of finding values for conversion rate and energy consumption that can be used for comparison purposes, taking into account each ADC type. Thus, regarding processing time, results show that the non-conventional sensor architecture requires fast ADCs, ideally one ADC per column, to report significant advantages. Regarding energy savings, the non-conventional architecture yields best results with SAR, cyclic or $\Sigma\Delta$ topologies. To reach that conclusion, we consider state-of-the-art experimental median figures regarding ADC energy consumption. Considering specific cases, the best case for energy savings is when the single-slope converter from [36] is used. By way of example, analysis using a column parallel SAR ADC with 14.6 pJ/sample shows that the architecture with pre-processing sensor can be 26 times faster and 49 times more energy-efficient than the digital approach with 10 PEs. The methodology presented in this paper allows for a quantitative estimation of the advantages that focal-plane processing might bring about. This is an interesting tool for imager designers to understand, before implementation, the strengths of the proposed focal-plane processing techniques.

Acknowledgments

This work was supported partly by Brazilian research funding agencies (CAPES, CNPq, and FAPERJ) through projects 309148/2013-8, 479437/2013-0, 309602/2016-5, E-26/201.514/2014 and grants 141288/2014-0, 204382/2014-9 and E-26/200.350/2016, partly by the Spanish Government through project TEC2015-66878-C3-1-R MINECO (European Region Development Fund, ERDF/FEDER), and partly by Junta de Andalucia through project TIC 2338-2013 CEICE and by the Office of Naval Research (USA) through grant N000141410355.

References

[1] A. Al-Fuqaha, M. Guizani *et al.*, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] R. Fontaine, "The state of the art of mainstream CMOS image sensors," in *Int. Image Sensor Workshop*, June 2015.

[3] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128×128 120 db 15μs latency asynchronous temporal contrast vision sensor," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 2, pp. 566–576, Feb 2008.

[4] T. Roska and A. Rodríguez-Vázquez, *Towards the Analogic Visual Microprocessor*. John Wiley & Sons, 2001.

[5] A. Zarándy, Ed., *Focal-plane Sensor-Processor Chips*. Springer, 2011.

[6] A. Rodríguez-Vázquez, R. C. Galán *et al.*, "In the quest of vision-sensors-on-chip: Pre-processing sensors for data reduction," in *IST Electronic Imaging*, Feb. 2017.

[7] E. Adelson, C. Anderson *et al.*, "Pyramid methods in image processing," *RCA Engineer*, vol. 29, no. 6, pp. 33–41, 1984.

[8] J. Campbell and V. Kazantsev, "Using an embedded vision processor to build an efficient object recognition system," *Synopsys - White Paper*, May 2015.

[9] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer-Verlag London Limited, 2010.

[10] T. Lindeberg, "Scale-space theory: A basic tool for analysing structures at different scales," *Journal of Applied Statistics*, vol. 21, no. 2, pp. 224–270, 1994.

[11] R. C. González and R. E. Woods, *Digital Image Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.

[12] J. Fernández-Berni, R. Carmona-Galán, and L. Carranza-González, "FLIP-Q: A QCIF resolution focal-plane array for low-power image processing," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 3, pp. 669–680, March 2011.

[13] M. Suárez, V. M. Brea *et al.*, "CMOS-3D smart imager architectures for feature detection," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 4, pp. 723–736, Dec 2012.

[14] E. R. Fossum and D. B. Hondongwa, "A review of the pinned photodiode for CCD and CMOS image sensors," *IEEE Journal of the Electron Devices Society*, vol. 2, no. 3, pp. 33–43, May 2014.

[15] "OpenVX," https://www.khronos.org/openvx/, accessed: 2017-31-03.

[16] H. Kobayashi, J. L. White, and A. A. Abidi, "An active resistor network for Gaussian filtering of images," *IEEE Journal of Solid-State Circuits*, vol. 26, no. 5, pp. 738–748, May 1991.

[17] Y. Ni, Y. M. Zhu *et al.*, "Yet another analog 2D Gaussian convolver," in *Circuits and Systems, 1993., ISCAS '93, 1993 IEEE International Symposium on*, May 1993, pp. 192–195 vol.1.

[18] L. Kabbai, A. Sghaiery *et al.*, "FPGA implementation of filtered image using 2D Gaussian filter," *Advanced Computer Science and Applications, International Journal of*, vol. 7, no. 7, 2016.

[19] B. Rajan and S. Ravi, "FPGA based hardware implementation of image filter with dynamic reconfiguration architecture," *Computer Science and Network Security, International Journal of*, vol. 6, no. 12, 2006.

[20] H. Zhang, M. Xia, and G. Hu, "A multiwindow partial buffering scheme for FPGA-based 2-D convolvers," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 2, pp. 200–204, Feb 2007.

[21] F. Oliveira, J. G. Gomes *et al.*, "Focal-plane scale space generation with a 6T pixel architecture," in *IS&T Electronic Imaging*, 2016.

[22] J. A. Leñero-Bardallo and A. Rodríguez-Vázquez, *ADCs for Image Sensors: Review and Performance Analysis*. CRC Press, 2016.

[23] K. Kiyoyama, K. W. Lee *et al.*, "A very low area ADC for 3-D stacked CMOS image processing system," in *3D Systems Integration Conference (3DIC), 2011 IEEE International*, Jan 2011, pp. 1–4.

[24] H. J. Kim, S. I. Hwang *et al.*, "Delta readout scheme for image-dependent power savings in a CMOS image sensor with multi-column-parallel SAR ADCs," in *2015 IEEE Asian Solid-State Circuits Conference (A-SSCC)*, Nov 2015, pp. 1–4.

[25] J. Y. Lin, K. H. Chang *et al.*, "An 8-bit column-shared SAR ADC for CMOS image sensor applications," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 301–304.

[26] M. K. Kim, S. K. Hong, and O. K. Kwon, "A small-area and energy-efficient 12-bit SA-ADC with residue sampling and digital calibration forCMOS image sensors," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 62, no. 10, pp. 932–936, Oct 2015.

[27] D. G. Chen, F. Tang, and A. Bermak, "A low-power pilot-DAC based column parallel 8b SAR ADC with forward error correction for CMOS image sensors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 10, pp. 2572–2583, Oct 2013.

[28] S. Matsuo, T. J. Bales *et al.*, "8.9-Megapixel video image sensor with 14-b column-parallel SA-ADC," *IEEE Transactions on Electron Devices*, vol. 56, no. 11, pp. 2380–2389, Nov 2009.

[29] D. G. Chen, F. Tang *et al.*, "A 64 fj/step 9-bit SAR ADC array with forward error correction and mixed-signal CDS for CMOS image

sensors," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 11, pp. 3085–3093, Nov 2014.

[30] M. S. Shin, J. B. Kim *et al.*, "A 1.92-Megapixel CMOS image sensor with column-parallel low-power and area-efficient SA-ADCs," *IEEE Transactions on Electron Devices*, vol. 59, no. 6, pp. 1693–1700, June 2012.

[31] R. Xu, W. C. Ng *et al.*, "A 1/2.5 inch VGA 400 fps CMOS image sensor with high sensitivity for machine vision," *IEEE Journal of Solid-State Circuits*, vol. 49, no. 10, pp. 2342–2351, Oct 2014.

[32] H. Le-Thai, A. Xhakoni, and G. Gielen, "A column-and-row-parallel CMOS image sensor with thermal and 1/f noise suppression techniques," in *ESSCIRC Conference 2016: 42nd European Solid-State Circuits Conference*, Sept 2016, pp. 221–224.

[33] A. Xhakoni, H. Le-Thai, and G. G. E. Gielen, "A low-noise high-frame-rate 1-D decoding readout architecture for stacked image sensors," *IEEE Sensors Journal*, vol. 14, no. 6, pp. 1966–1973, June 2014.

[34] Y. Chae, J. Cheon *et al.*, "A 2.1 M pixels, 120 frame/s CMOS image sensor with column-parallel $\Delta\Sigma$ ADC architecture," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 1, pp. 236–247, Jan 2011.

[35] Y. Oike and A. E. Gamal, "CMOS image sensor with per-column $\Sigma\Delta$ ADC and programmable compressed sensing," *IEEE Journal of Solid-State Circuits*, vol. 48, no. 1, pp. 318–328, Jan 2013.

[36] Y. Oike, K. Akiyama *et al.*, "An 8.3M-pixel 480fps global-shutter CMOS image sensor with gain-adaptive column ADCs and 2-on-1 stacked device structure," in *2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits)*, June 2016, pp. 1–2.

[37] A. Spivak, A. Belenky, and O. Yadid-Pecht, "Very sensitive low-noise active-reset CMOS image sensor with in-pixel ADC," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 10, pp. 939–943, Oct 2016.

[38] Y. Lim, K. Koh *et al.*, "A 1.1e- temporal noise 1/3.2-inch 8Mpixel CMOS image sensor using pseudo-multiple sampling," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, Feb 2010, pp. 396–397.

[39] S. Kleinfelder, S. Lim *et al.*, "A 10000 frames/s CMOS digital pixel sensor," *IEEE Journal of Solid-State Circuits*, vol. 36, no. 12, pp. 2049–2059, Dec 2001.

[40] M. F. Snoeij, A. J. P. Theuwissen *et al.*, "Multiple-ramp column-parallel ADC architectures for CMOS image sensors," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 12, pp. 2968–2977, Dec 2007.

[41] S. Lim, J. Lee *et al.*, "A high-speed CMOS image sensor with column-parallel two-step single-slope ADCs," *IEEE Transactions on Electron Devices*, vol. 56, no. 3, pp. 393–398, March 2009.

[42] T. Toyama, K. Mishina *et al.*, "A 17.7Mpixel 120fps CMOS image sensor with 34.8Gb/s readout," in *2011 IEEE International Solid-State Circuits Conference*, Feb 2011, pp. 420–422.

[43] Y. Nitta, Y. Muramatsu *et al.*, "High-speed digital double sampling with analog CDS on column parallel ADC architecture for low-noise active pixel sensor," in *2006 IEEE International Solid State Circuits Conference - Digest of Technical Papers*, Feb 2006, pp. 2024–2031.

[44] J. Lee, H. Park *et al.*, "High frame-rate VGA CMOS image sensor using non-memory capacitor two-step single-slope ADCs," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 9, pp. 2147–2155, Sept 2015.

[45] T. Lyu, S. Yao *et al.*, "A 12-bit high-speed column-parallel two-step single-slope analog-to-digital converter (ADC) for CMOS image sensors," *Sensors*, no. 14, 2014.

[46] J. Bae, D. Kim *et al.*, "A two-step A/D conversion and column self-calibration technique for low noise CMOS image sensors," *Sensors*, no. 14, 2014.

[47] F. Tang, B. Wang *et al.*, "A column-parallel inverter-based cyclic ADC for CMOS image sensor with capacitance and clock scaling," *IEEE Transactions on Electron Devices*, vol. 63, no. 1, pp. 162–167, Jan 2016.

[48] J. H. Park, S. Aoyama *et al.*, "A high-speed low-noise CMOS image sensor with 13-b column-parallel single-ended cyclic ADCs," *IEEE Transactions on Electron Devices*, vol. 56, no. 11, pp. 2414–2422, Nov 2009.

[49] M. Mase, S. Kawahito *et al.*, "A wide dynamic range CMOS image sensor with multiple exposure-time signal outputs and 12-bit column-parallel cyclic A/D converters," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 12, pp. 2787–2795, Dec 2005.

[50] S. Lim, J. Cheon *et al.*, "A 240-frames/s 2.1-mpixel CMOS image sensor with column-shared cyclic ADCs," *IEEE Journal of Solid-State Circuits*, vol. 46, no. 9, pp. 2073–2083, Sept 2011.

[51] K. Kitamura, T. Watabe *et al.*, "A 33-Megapixel 120-frames-per-second 2.5-Watt CMOS image sensor with column-parallel two-stage cyclic analog-to-digital converters," *IEEE Transactions on Electron Devices*, vol. 59, no. 12, pp. 3426–3433, Dec 2012.

[52] M. Furuta, Y. Nishikawa *et al.*, "A high-speed, high-sensitivity digital CMOS image sensor with a global shutter and 12-bit column-parallel cyclic A/D converters," *IEEE Journal of Solid-State Circuits*, vol. 42, no. 4, pp. 766–774, April 2007.

[53] J. H. Park, S. Aoyama *et al.*, "A high-speed low-noise cis with 12b 2-stage pipelined cyclic ADCs," in *2011 International Image Sensor Workshop*, Jun 2011.

[54] M. H. Choi, G. C. Ahn, and S. H. Lee, "12b 50 MS/s 0.18 μm CMOS ADC with highly linear input variable gain amplifier," *Electronics Letters*, vol. 46, no. 18, pp. 1254–1256, September 2010.

[55] S.-H. Cho, J.-S. Park *et al.*, "A 14—10 B dual-mode low-noise pipeline ADC for high-end CMOS image sensors," *Analog Integr. Circuits Signal Process.*, vol. 80, no. 3, pp. 437–447, Sep. 2014.

[56] S. Zhang, L. Xiaokang *et al.*, "A 12-bit 96Msample/s double-data-rate (DDR) pipeline ADC with speed and noise optimization for CMOS image sensors," in *2014 International Conference on Information Science, Electronics and Electrical Engineering*, vol. 3, April 2014, pp. 1798–1803.

[57] J. S. Park, T. J. An *et al.*, "A 10b 50MS/s 90nm CMOS skinny-shape ADC using variable references for CIS applications," in *2013 International SoC Design Conference (ISOCC)*, Nov 2013, pp. 080–082.

[58] K. B. Cho, C. Lee *et al.*, "A 1/2.5 inch 8.1Mpixel CMOS image sensor for digital cameras," in *2007 IEEE International Solid-State Circuits Conference. Digest of Technical Papers*, Feb 2007, pp. 508–618.

[59] B. Murmann, "Trends in low power, digitally assisted A/D conversion," *IEICE Transactions on Electronics*, vol. 93, no. C(6), pp. 718–729, June 2010.

[60] M. Pelgrom, *Analog-to-Digital Conversion*, 1st ed. Springer, 2010.

[61] G. J. Sullivan, J. R. Ohm *et al.*, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec 2012.

[62] R. H. Walden, "Analog-to-digital converter survey and analysis," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 4, pp. 539–550, Apr 1999.

[63] M. ElDesouki, M. Jamal Deen *et al.*, "CMOS image sensors for high speed applications," *Sensors*, vol. 9, no. 1, 2009.

[64] B. E. Jonsson, "An empirical approach to finding energy efficient ADC architectures," in *2011 International Workshop on ADC Modelling, Testing and Data Converter Analysis and Design and IEEE 2011 ADC Forum*, July 2011.

[65] J. Rabaey, *Low Power Design Essentials*. 233 Spring Street, New York, NY 10013, USA: Springer, 2009.

[66] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[67] "Affine covariant features," http://www.robots.ox.ac.uk/ vgg/research/affine/, accessed: 2016-11-02.

Fernanda D. V. R Oliveira graduated as an Electronic Engineer in 2012 and by the end of 2013 received her M.S. degree in Electric Engineering, both from the Federal University of Rio de Janeiro (UFRJ), Brazil. She is currently pursuing her Ph.D. degree in microelectronics at the Electrical Engineering Program of COPPE/UFRJ. She did a one year internship in the Microelectronics Institute of Seville in 2015. Her research fields are image sensors and image processing.

José Gabriel R. C. Gomes graduated in Electrical Engineering from the Federal University of Rio de Janeiro in 1999 (magna cum laude). He obtained M.S. degrees in Electrical Engineering from COPPE/UFRJ (2000) and from the University of California at Santa Barbara (UCSB, 2003), and a Ph.D. degree in Electrical Engineering from UCSB (2004). In 2005, he was a post-doctoral researcher with the Electrical Engineering Program at COPPE/UFRJ. In 2006, he joined the faculty of the Electronics and Computer Engineering Department at the Federal University of Rio de Janeiro, where he is currently an Associate Professor. Since 2007, he is also part of the faculty at the Electrical Engineering Program at COPPE/UFRJ. His professional experience concentrates on Electronics Instrumentation, with an emphasis on CMOS image sensors, image compression, and neural networks. He received "Young Researcher of Our State" research grants from FAPERJ/Brazil for terms 2009/2012 and 2015/2017. He and his co-authors received the Best Paper Award at the 25th Symposium on Integrated Circuits and Systems Design (SBCCI 2012) in Brasilia, Brazil. He was a recipient of the IEEE Circuits and Systems Society Chapter-of-the-Year Award (Region 9, 2009). He is a IEEE member since 2001.

Jorge Fernández-Berni received a B. Eng. degree in Electronics and Telecommunication in September 2004, a M.Sc. degree in Microelectronics in December 2008 and his Ph.D. in June 2011 with honors, from the University of Seville, Spain. From January 2005 through September 2006, he was working in the Telecommunication Industry. He has been a visiting researcher at the Computer and Automation Research Institute (Budapest, Hungary), Ghent University (Ghent, Belgium) and the University of Notre-Dame (IN, USA). Dr. Fernández-Berni has authored/co-authored some 50 papers in refereed journals, conferences and workshops. He is also the first author of a book and two book chapters as well as the first inventor of two licensed patents. He received the Best Paper Award in "Image Sensors and Imaging Systems, SPIE Electronic Imaging 2014, San Francisco CA, USA" and the Third Prize of the Student Paper Award in "IEEE CNNA 2010: 12th Int. Workshop on Cellular Nanoscale Networks and their Applications, Berkeley CA, USA". His main areas of interest are smart image sensors, vision chips and embedded vision systems.

Ricardo Carmona-Galán graduated in Electronic Physics and got a Ph.D. in Microelectronics from the University of Seville, Spain. He worked as a Research Assistant at the EECS Department of the University of California, Berkeley. He was Assistant Professor at the School of Engineering of the University of Seville. Since 2005, he is a Tenured Scientist at the Institute of Microelectronics of Seville (CSIC-Univ. Seville). He also held a Postdoc at the University of Notre Dame, Indiana, where he worked in interfaces for CMOS compatible nanostructures for multispectral light sensing. His main research focus has been on VLSI implementation of concurrent sensor/processor arrays for real time image processing and vision. He has designed several vision chips implementing different focal plane operators for early vision processing. His current research interests lie in the design of low-power smart image sensors, single-photon detection and ToF estimation, and 3-D integrated circuits for autonomous vision systems. He has coauthored more than 120 journal and conference papers and a book on low-power vision sensors for vision-enabled sensor networks. He is co-inventor of several patents. He has collaborated with start-up companies in Seville (Anafocus) and Berkeley (Eutecus). Ricardo Carmona-Galán is a Senior Member of the IEEE and member of the IEEE Circuits and Systems and Solid-State Circuits Societies. He has been associate editor for IEEE TCAS-I and now is for Springer's Journal on Real-Time Image Processing. He got a Certificate of Teaching Excellence from the University of Seville. Very recently, he has received the Best Paper Award of the IEEE-CASS Technical Committee on Sensory Systems at ISCAS 2016, together with Dr. I. Vornicu and Prof. A. Rodríguez-Vázquez.

Rocío del Río received the M.S. degree in 1996 in Electronic Physics and the Ph.D. degree in Microelectronics in 2004, both from the University of Seville, Spain. She joined the Department of Electronics and Electromagnetism of the University of Seville in 1995, where she is an Associate Professor. She is also since 1995 at the Institute of Microelectronics of Seville (IMSE-CNM, CSIC/University of Seville), where she works in the group of "Analog and Mixed-Signal Microelectronics". Her main research interests are in the field of mixed-signal circuits (with special emphasis in switched-capacitor circuit techniques) and analog-to-digital converters, including analysis, behavioral modeling, and design automation (specially, of $\Sigma\Delta$ ADCs). She has participated in diverse National and International R&D projects and has co-authored more than 100 international publications, including journal and conference papers, books chapters, and the books *CMOS Cascade Sigma-Delta Modulators for Sensor and Telecom: Error Analysis and Practical Design* (Springer, 2006), *Nanometer CMOS Sigma-Delta Modulators for Software Defined Radio* (Springer, 2011), and *CMOS Sigma-Delta Converters: Practical Design Guide* (Wiley-IEEE Press, 2013).

Ángel Rodríguez-Vázquez (IEEE Fellow, 1999) Ángel Rodríguez-Vázquez (F'99) received the bachelor's (Universidad de Sevilla, 1976) and Ph.D. degrees in physics-electronics (Universidad de Sevilla, 1982) with several national and international awards, including the IEEE Rogelio Segovia Torres Award (1981). After different research stays in University of California-Berkeley and Texas A&M University he became a Full Professor of Electronics at the University of Sevilla in 1995. He co-founded the Instituto de Microelectrónica de Sevilla, a joint undertaken of the Consejo Superior de Investigaciones Científicas (CSIC) and the University of Sevilla and started a Resaerch Lab on Analog and Mixed-Signal Circuits for Sensors and Communications. In 2001 he was the main promotor and co-founder of the start-up company AnaFocus Ltd and served as CEO, on leave from the University, until June 2009, when the company reached maturity as a worldwide provider of smart CMOS imagers and vision systems-on-chip. His research is on the design of analog and mixed-signal front-ends for sensing and communication, including smart imagers, vision chips, implantable neural recorders/stimulators and biomedical circuits and with emphasis on system integration. He has authored 11 books, 36 additional book chapters, and some 500 articles in peer-review specialized publications. He has presented invited plenary lectures at different international conferences. His research work has received some 8,184 citations; he has an h-index of 46 and an i10-index of 165. Dr. Rodríguez-Vázquez has received a number of awards for his research (the IEEE Guillemin-Cauer Best Paper Award, two Wiley's IJCTA Best Paper Awards, two IEEE ECCTD Best Paper Awards, one IEEE-ISCAS Best Paper Award, one SPIE-IST Electronic Imaging Best Paper Award, the IEEE ISCAS Best Demo-Paper Award, and the IEEE ICECS Best Demo-Paper Award). Prof. Rodríguez-Vázquez has always been looking for the balance between long-term research and innovative industrial developments. He was the main promotor and co-founder of AnaFocus Ltd. and he participated in the foundation of the Hungarian start-up company AnaLogic Ltd. He has eight patents filed, some of which are licensed to companies. AnaFocus was founded on the basis of his patents on vision chip architectures. He has served as Editor, Associate Editor, and Guest Editor for IEEE and non-IEEE journals, is on the committee of several international journals and conferences, and has chaired several international IEEE and SPIE conferences. He served as VP Region 8 of the IEEE Circuits 1087 and Systems Society (2009-2012) and as Chair of the IEEE CASS Fellow Evaluation Committee (2010, 2012, 2013, 2014, and 2015).