FLIP-Q: A QCIF Resolution Focal-Plane Array for Low-Power Image Processing

Jorge Fernández-Berni, Ricardo Carmona-Galán, *Associate Member, IEEE* and Luis Carranza-González

Abstract

This paper reports a 176×144-pixel smart image sensor designed and fabricated in a 0.35μm CMOS-OPTO process. The chip implements a massively parallel focal-plane processing array which can output different simplified representations of the scene at very low power. The array is composed of pixel-level processing elements which carry out analog image processing concurrently with photosensing. These processing elements can be grouped into fully-programmable rectangular-shape areas by loading the appropiate interconnection patterns into the registers at the edge of the array. The targeted processing can be thus performed block-wise. Readout is done pixel-by-pixel in a random access fashion. On-chip 8b ADC is provided. The image processing primitives implemented by the chip, experimentally tested and fully functional, are scale space and Gaussian pyramid generation, fully-programmable multiresolution scene representation — including foveation — and block-wise energy-based scene representation. The power consumption associated to the capture, processing and A/D conversion of an image flow at 30fps, with full-frame processing but reduced frame size output, ranges from 2.7mW to 5.6mW, depending on the operation to be performed.

Manuscript received ???; revised ???. This work was supported by CICE/JA and MICINN (Spain) through projects 2006-TIC-2352 and TEC 2009-11812, co-funded by FEDER, respectively.

The authors are with the Institute of Microelectronics of Seville (IMSE-CNM), Consejo Superior de Investigaciones Científicas y Universidad de Sevilla, C/ Américo Vespucio s/n, 41092, Seville, Spain (e-mail: berni@imse-cnm.csic.es).

Index Terms

CMOS image sensors, low-power image processing, focal-plane scale space, simplified scene representation, multiresolution, foveation

I. INTRODUCTION

Image processing is usually divided into three consecutive steps: i) low-level tasks, where both inputs and outputs are images, ii) medium-level tasks, where inputs are images but outputs are attributes extracted from inputs and iii) high-level tasks, which perform the cognitive functions associated to vision from the result of low- and medium-level tasks. The main feature of low-level tasks is their intrinsic parallelism as they are equally defined for each pixel, usually as a function of its own and its immediate neighborhood's value. This makes the conventional imager-memory-DSP architecture rather unsuited to carry out low-level image processing. The speed of the digital processor must be high in order to handle the massive data flow and the repeated memory accesses, what drastically affects the power consumption [1]. Alternative architectures can be proposed to handle low-level image processing tasks more efficiently. They can take advantage of the moderate accuracy usually required in early vision models [2]. For instance, instead of delivering the already captured raw data, a simplified representation of the scene can be elaborated with relatively coarse circuitry at the focal plane. Higher level vision tasks can be implemented then by conventional digital architectures, now operating on a reduced processing load, and consequently lowering the overall power consumption.

This architectural scheme has been incorporated to either general-purpose vision chips [3], [4] or to application-specific smart image sensors [5], [6]. Mainly thanks to the ability of CMOS processes, unlike CCD technology, to integrate imaging with signal processing. The basic idea behind these approaches is to incorporate a processing element (PE) next to the photosensor at every pixel, obtaining thus focal-plane processing arrays. In such arrays, the Single

Instruction Multiple Data (SIMD) paradigm, sketched back in 1958 [7], is usually applied. All the PEs execute the same instructions while making computations on different data. Although SIMD-based focal-plane processing arrays composed of digital PEs have been proposed [8], [9], currently the analog implementations continue to be more area- and power-efficient than their digital counterparts.

In this paper, we present a vision chip intended for applications with really strict power budgets [10], [11]. It is based on a focal-plane processing array comprised of analog PEs. These PEs exploit the large signal behaviour of the transistors in order to achieve very high efficiency in terms of both area and power consumption. The image processing primitives implemented permits enough flexibility to generate different degrees of simplification of the scene according to the requirements of the vision algorithm. These primitives are:

- *Progressive spatial filtering and subsequent subsampling.* It leads to scale space and Gaussian pyramid generation [12], [13], permitting image analysis on the desired spatial frequencies. The chip can perform this operation over rectangular-shape user-defined subimages.
- *Fully-programmable multiresolution scene representation.* Different resolutions can be obtained by grouping pixels in rectangular-shape user-defined blocks. Progressive coarse-to-fine resolutions can be also programmed in order to achieve foveation, that is, to keep full resolution only in regions of interest (ROI) within the scene.
- *Block-wise energy-based scene representation.* This primitive, along with the progressive spatial filtering, permits to efficiently segment spatially-repetitive patterns and high contrast zones at different scales within the scene.

All these primitives have been tested and are fully functional in a chip manufactured in the AMS 0.35μm CMOS-OPTO process. This CMOS process does not incorporate any special device for image sensing. Indeed, it only differs from the standard AMS 0.35μm process in an

anti-reflective coating and an EPI substrate which reduces the dark current. The chip contains around half million transistors, 98% of them working in analog mode.

II. Architecture

The architecture of the chip is depicted in Fig. 1. The analog core is a 176×144 array of PEs with concurrent photodiodes. The PEs are 4-connected. Each of these connections can be enabled or disabled column-wise and row-wise across the array. The focal plane can be divided into independent rectangular blocks whose size is defined by the user by selecting which columns and rows of PEs are interconnected. Note that the size of the blocks could vary across the focal plane. Once this block-based division is set, the control logic for diffusion and/or energy computation generates the corresponding signals to perform any of the processing primitives mentioned in Section I. All the circuitry and signals involved until eventually carrying out a certain primitive are detailed in Section III.

The outcome of the processing can be read out pixel-wise by selecting the column and row where the desired pixel is located. The value of the pixel is buffered at the column bus and delivered to a 8b SAR ADC, which finally outputs the digitalized result. Although the inclusion of only one ADC prevents the chip from reaching high frame rates as a full-resolution imager, it greatly reduces the power consumption while still allowing a remarkable throughput for the simplified representations of the scene achievable at the focal plane.

The main characteristics of the chip are summarized in Table I. A microphotograph with a close-up of the photosensors is shown in Fig. 2. Experimental results are reported in Section IV along with a comparison to other chips in the literature.

III. Image processing implementation

A. *Diffusion-based filtering*

The elementary cell of the analog core is depicted in Fig. 3, and a timing diagram with the control signals and the waveform of the voltage at the most relevant nodes of the basic cell is shown in Fig. 4. The nominal reset voltage of the photodiode and the sensing capacitance C_P is 2.5V. It can be extended to 3.3V though accuracy of the analog blocks is compromised because the MOS-based resistors become more nonlinear. The control signals 'rst' and 'read' implement an electronic global shutter (see Fig. 4). The analog pixel value is represented by the voltage V_{ij} after integration time. C_P is 4-connected to its neighbors through MOS-based resistors, implementing a MOS-based RC network. Note that each linking MOS-resistor is shared with the corresponding neighbor cell. The equivalent resistance R_{eq} of these transistors, tailored as reported in [14], along with the value of the C_P determine the time constant of the network $\tau = R_{eq}C_P$. As it will be more evident later, the value of the time constant is related with our ability to control the duration of the diffusion. We have implemented an internal VCO to clock finer steps in the duration of the diffusion. Frequencies up to 150MHz can be implemented. Correspondingly, the smallest diffusion step, t_{min}, will be 6-7ns. On the other hand, as a system specification, we considered that Gaussian filters with widths below $\sigma = 1$ must be achieved. Thus, really gradual scale spaces can be generated. Since $\sigma = \sqrt{2t/\tau}$, as explained shortly, a value of τ around one order of magnitude greater than t_{min} is enough to fulfill this system specification by far. Nominal τ was decided to be 85ns, granting margin of error to the maximum frequency reachable by the VCO. With this value, the design procedure starts by selecting the value of the capacitor C_P. As it is the sensing capacitance, a trade-off between sensitivity and the minimization of the reset error leads to $C_P = 1$pF. Then, continues with an automatic search for a transistor implementing 85kΩ. The initial guess for the design of the corresponding transistor

is given by the formula in [14].

The key aspect from the point of view of the image processing is that the connection between any two neighbor nodes can be controlled through the gate voltage of the transistor which links them, namely signals $S_{C_{i-1,i}}$, $S_{C_{i,i+1}}$, $S_{R_{j-1,j}}$ and $S_{R_{j,j+1}}$ in Fig. 4. When off, the corresponding nodes are disconnected. When on, the linking transistor behaves as a resistor of value R_{eq}. This control, performed column-wise and row-wise, has two objectives. First of all, a permanent disconnection between certain consecutive columns and rows across the array determines the boundaries of the blocks in which the focal plane is divided. Secondly, a time-controlled connection between consecutive columns and rows implements a spatially-discretized diffusion process over the voltages V_{ij} within the respective blocks. The equation which defines this process is:

$$\tau \frac{dV_{ij}}{dt} = -4V_{ij} + V_{i+1,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} \tag{1}$$

whose solution is formally the scale-space representation of 2-D discrete signals [12]. Notice that the dynamics of those cells located just at the edge of a block is not determined by a complete 4-connected neighborhood but by a reduced 2- or 3-connected one. It is equivalent to consider mirroring boundary conditions at every time instant for the edges of every block. By applying the DFT to Eq. (1) and solving in time, we obtain the following transfer function:

$$\hat{H}_{uv}(t) = \frac{\hat{V}_{uv}(t)}{\hat{V}_{uv}(0)} = e^{-\frac{4t}{\tau}\left[\sin^2\left(\frac{\pi u}{W}\right)+\sin^2\left(\frac{\pi v}{H}\right)\right]} \tag{2}$$

where $\hat{V}_{uv}(0)$ represents the DFT of a $W \times H$ block defined by the corresponding voltages V_{ij} just after capturing a new frame and $\hat{V}_{uv}(t)$ is the DFT of the same block defined by the voltages V_{ij} after letting the charge stored in C_P diffuse for a time t. This transfer function approximates

a continuous-plane Gaussian spatial filter with $\sigma = \sqrt{2t/\tau}$. The scale parameter associated to the scale-space representation is defined as:

$$\xi = \sigma^2 = 2\frac{t}{\tau} \tag{3}$$

Therefore, thanks to the MOS-based RC network, it is possible to generate a scale space within user-defined divisions of a scene. This reconfigurable operation entails crucial advantages from the point of view of simplifying the representation of a scene. Firstly, Gaussian pyramids can be easily built by subsampling each image of the scale space according to the scale [13]. It permits to directly extract from the focal-plane processing a representation of the scene containing only the spatial frequencies of interest. Secondly, notice from Eq. (2) that a long enough diffusion ($t \rightarrow \infty$) filters all the spatial frequencies except the dc component. It means that the final value of all the pixels after a complete diffusion will be the average of their initial values. This property, along with the reconfigurability of the array, permits to achieve fully-programmable multiresolution representations of a scene by binning pixels.

In order to achieve a fine control of the diffusion time, i. e. fine-grain selection of the spatial bandwidth of the filtering performed by the network, a diffusion control module, common to the pixels array, is implemented. Its main component is a 12b shift register (SHR) that shapes the diffusion control signal driving the MOS-resistor gates. In order to provide some guard time for internal timing of the operation, the first two bits introduced into the SHR, that is, the first two bits defining the diffusion time, must be set to zero. Thus, only 10 bits are effectively employed to define the pulse duration. An external clock or an internal VCO can be employed to shift the register. The combined effect of two parameters, N_1, which is the number of logic '1's stored in the SHR and f_{CLK}, the frequency of the clock, leads to a diffusion time given by:

$$t = \frac{N_1}{f_{CLK}} \tag{4}$$

If the internal VCO selected, the parameter f_{CLK} depends in turn on a fine adjustment of its control voltage. The VCO is a fifteen-stage ring oscillator based on pseudo-NMOS inverters whose load current is controlled by an external biasing signal in order to vary the propagation delay of each stage and, consequently, the frequency. Frequencies ranging from 0.5MHz to around 150 MHz can be attained. It means that t could ideally take any value within the interval $[6.66ns, 20us]$ by simply realizing a fine setting of N_1 and f_{CLK}. The minimum value of t is around one order of magnitude smaller than τ. It entails the possibility of generating really fine scales since the scale parameter depends on the quotient t/τ, as pointed out in Eq. (3).

B. Image energy computation

The progressive spatial filtering performed during the generation of the scale space also allows for further simplified scene representations. Let $V_{ij}(t)$ be the voltages at the nodes of a $W \times H$ block after a certain interval of diffusion t. The total energy of the block is defined as:

$$E(t) = \sum_{i=1}^{W} \sum_{j=1}^{H} |V_{ij}(t)|^2 = \sum_{u=0}^{W-1} \sum_{v=0}^{H-1} |\hat{V}_{uv}(t)|^2 \tag{5}$$

Eq. (5) along with Eq. (2) imply that the amount of energy that remains in the block accounts for the filtering undergone during the diffusion. In other words, the energy at each time instant is a measure of the evolution of the diffusion process. The longer t the less $E(t)$. The energy lost between two consecutive points in time during the difussion corresponds to that of the spatial frequencies filtered. In this way, the single value of the energy along the scale space summarizes the frequency content of the block. In order to efficiently compute the block energy at the focal plane, we are making use of the MOS transistor square law and the summation

of the contribution of the individual pixels in the form of currents. It is implemented by the transistor M_E (Fig. 3), working in saturation, the capacitor C_E, the switches S_E and S_{pre} and MOS switches for charge redistribution which average the voltages $V_{E_{ij}}$ within the block. Firstly, as we are interested in the computation of the energy associated to the previously defined blocks of the image, the same block division as for the voltages $V_{ij}(t)$ is established by the selection signals $S_{CE_{m,m+1}}$ and $S_{RE_{n,n+1}}$. Then, all the capacitors C_E are precharged to V_{DD}, 3.3V, by switching on both S_E and S_{pre} (see this happening twice, one before diffusion and one after diffusion, in the diagram of Fig. 4). Then, S_{pre} is switched off while S_E is kept on during a time interval T_E, 20ns in our case, discharging C_E through M_E. Once S_E is definitely switched back off, the voltage at C_E would be, with respect to V_{DD}, proportional to the pixel energy:

$$V_{E_{ij}} = V_{DD} - \frac{T_E}{C_E}\beta[V_{ij}(t) - V_{th}]^2 \tag{6}$$

where V_{th} is the threshold voltage and β the transconductance parameter of M_E. However, due to the charge redistribution realized through the MOS resistors, the following value is eventually reached:

$$V_{E_{ij}} = V_{DD} - \frac{\beta T_E}{WHC_E}\sum_{i=1}^{W}\sum_{j=1}^{H}[V_{ij}(t) - V_{th}]^2 \tag{7}$$

which is, again with respect to V_{DD}, proportional to the total energy of that block t seconds after the diffusion started. In the ideal case in which all the M_E transistors perfectly match, the offset introduced by V_{th} will not affect the computation of the energy associated to any spatial frequency other than the dc component. In the real chip, V_{th} is subject to across die variations, as are other transistor parameters. This induces FPN to appear. We have measured the amount of FPN present in the energy representation of each individual pixel. First, while keeping the capacitors C_E on reset, the output is sampled several times in order to filter out the temporal

noise contribution. The result is a a standard deviation of 1.12% referred to the full signal range of the output corresponding to the readout of the energy representation. In addition, we have allowed the capacitors to discharge for a uniform image in the middle of the range, i. e. 2.0V at node V_{ij}, also for a number of times. Subtracting the averaged values obtained before from these later ones the standard deviation is now 7.85%. This value summarizes the contribution of the mismatch of V_{th}, amplified by the transistor square-law, the mismatch in the transconductance of M_E, and the switching errors introduced by S_E and S_{pre}, because they need to be switched for the computation of the energy and C_E is not as large as C_P. However, this computation is hardly applied to individual pixels. It is usually employed to represent the energy content of a group of pixels. This constitutes a spatial lowpass filter that reduces the influence of FPN. In order to achieve the reduced representation of the scene, only one pixel out of every block needs to be read as all the capacitors within the block will be at the same voltage defined by Eq. (7).

This simplified representation of the scene makes possible to efficiently segment spatially-repetitive patterns by monitoring the value of the energy along the scale space. Besides, the difference between the initial value of the energy and the energy after a complete diffusion (t long enough) accounts for the contrast within the block considered. The more the value of this difference, the more the intensity changes which determine the frequency content of the block. This information allows for a first estimation of the salient regions of the scene [15].

C. Block division control logic

It comprises the column-block and row-block control logic modules in Fig. 1. These modules generate the appropiate selection signals to configure the image sub-blocks. Links between cells within the same block are enabled. Disabling a column/row across the array establishes one of the boundaries of the adjacent blocks. We are going to focus on the column-block control logic (Fig. 5) as its description is directly applicable to the row-block control logic. The operation

is based on a SHR which is externally loaded and clocked. Each bit of the register determines the link between two columns of PEs. Thus the bit 'i' storing a logic value '1' determines that columns 'i' and 'i+1' are linked. On the contrary, the bit 'i' storing a logic value '0' establishes that columns 'i' and 'i+1' are unlinked. This scheme allows for an easy and fast reconfiguration of the blocks by adequately shifting the patterns loaded into the registers. Besides, it is specially suited for a microcontroller as only four pins — two for the column register and two for the row register — suffice to define the focal-plane division. The internal, active-high, signal $diff\_ctrl$ comes from the diffusion control logic. This signal controls the time interval t of diffusion filtering within the blocks once the focal-plane division is set. The signal 'energ_en' enables in turn the computation of the block energy. Notice that each and every signal $S_{C_{m,m+1}}$ and $S_{CE_{m,m+1}}$ — correspondingly $S_{R_{n,n+1}}$ and $S_{RE_{n,n+1}}$ in the row-control logic — must be buffered in order to achieve an accurate timing of the control logic across the array. It benefits the accuracy of the processing. In fact, all the signals which must nominally reach the whole array at the same time are carefully buffered.

IV. Experimental results

A. Calibration of the time constant for diffusion

The nominal value of the time constant for the time-controlled diffusion at the focal plane is $\tau = 85$ns, as mentioned in Section III-A. The value of τ is the product of a capacitance and a resistance, both implemented by MOS transistors. Within the same chip, mismatch from one pixel to another can be reduced by selecting large area devices (Fig. 3). Simulation using deviation parameters provided by the foundry is employed to confirm the minimization of the effect [16]. During the test of the chip no significant signs of anisotropy in the diffusion, due to time constant mismatch, has been appreciated. However, the value of τ is quite sensitive to process parameter deviations from chip to chip. It is therefore necessary, for the characterization

of the chip operation, a calibration process in order to determine its actual value in the sample under test. The target of the calibration of τ is to obtain an experimental value that can be employed off-chip to generate the response of an ideal RC network. If the nominal value of the time constant is employed instead of the measured τ, the response of the chip will greatly deviate from the ideal response. In order to disaggregate errors due to other causes, the actual τ implemented by the chip needs to be measured. With this value, the actual bandwidth of the implemented Gaussian filter can be precisely determined, and thus the goodness of the approximation can be established.

The calibration process consists in measuring the evolution of the voltage at two coupled pixels whose initial voltages can be externally set. There is a pair of accessible pixels at each side of the array, in order to take the across-die variations into account. Before testing any dynamic magnitude, each pixel's source follower is characterized in order to extract deviations introduced by the buffer from node V_{ij} measurements. For each pair, the initial voltages are set to V_{min} and V_{max} and then diffusion is allowed to evolve. As demonstrated in [14], the resistance R_{eq} best emulated by the MOS resistor is its instantaneous resistance when the sum of the drain and source voltages equals $V_{min} + V_{max}$. An ideal diffusion between a node set to V_{min} and another one set to V_{max} meets this at every time instant. Having meaured enough points within the close-to-exponential decay of the nodes, a least square fitting of these points with ideal exponential curves varying τ is realized. The result for the upper left corner is depicted in Fig. 6. Here, the evolution of the voltages V_{11} (Chip pixel 1) and V_{12} (Chip pixel 2) are compared with the evolution of the corresponding nodes of an ideal network (Ideal pixels 1 and 2) implementing the τ obtained in the error minimization, i. e. $\tau = 72.4$ns. A RMSE of 2.26% is obtained for this τ. In the upper right corner, a minimum RMSE of 0.58% is reached for $\tau = 69.8$ns. These values make perfect sense taking into account that, according to simulations at the corners of the technology, τ can range from 49ns (WP corner) to 148ns (WS corner). The value of τ that will

be employed for the comparisons from now on in the text will be the average of the extracted values, that is, $\tau = 71.1$ns.

B. Scale space

Once τ is calibrated, any on-chip scale space can be compared to its ideal counterpart obtained by solving the spatially-discretized diffusion equation. A single image is captured to be the initial image of both the on-chip scale space and the ideal scale space calculated off-chip. This capture is affected by a 0.72% FPN. It has been calculated by averaging a set of readings of the whole array without photocurrent integration, in order to skip temporal fluctuations, and then computing the standard deviation. No FPN removal circuit is included in the chip, neither is performed off-chip. Back to the scale space, the on-chip scale space is generated by applying successive diffusion steps to the original captured image. After every step, the image is converted to digital and delivered to the test instruments to be compared to the ideal image generated by MATLAB® in terms of the RMSE (Fig. 7). Some of the diffusion steps are represented in Fig. 8 (first row) and compared to the ideal images (second row). The last row contains a pictorial representation of the error, normalized in each case to the highest measured error on an individual pixels, which are 0%, 24.99%, 19.39%, 6.17%, 3.58% and 6.68%, respectively. Note that these large errors on certain pixels have little qualitative effect over the images. It can also be seen how noise eventually becomes dominant at coarse scales. Keep in mind that readout noise is present at the initial image of both scale spaces, but it is only added to each subsequent image of the on-chip scale space because of the readout mechanism. It means that while the initially stored noise, spatial and temporal, is progressively averaged in the ideal scale space, it is resampled for each picture of the on-chip scale space. As a consequence, there is an increase in the error for a sufficiently large diffusion duration. The key point here is that the accuracy of the processing predicted by simulation [14] is very close to that of the first images of the scale space, where

noise is not dominant yet. Besides, the error is kept under a reasonable level despite no FPN removal is carried out. This fact together with the efficiency of the focal-plane operation is crucial for artificial vision applications under strict power budgets.

C. Gaussian pyramids

Scale-space representations successively become more redundant as the scale parameter increases. A progressive filtering is performed over the scene, starting from the highest spatial frequencies and continuing until eventually filtering all the frequencies other than the dc component. However, in this process, the resolution of the images does not change and the oversampling of the remaining frequency content constantly increases along the scale space. Pyramid representations solve this problem by subsampling the scale-space representations according to the filtering realized. The control flow for this operation is simple: (1) after image capture, the diffusion time is set to match the required scale; (2) diffusion is realized; (3) the resulting image is subsampled at the appropriate rate, 2, 4, etc.; (4) go back to (1) and set the diffusion time to match the following scale, but taking into account that the stored image is already filtered. As an example, consider the scale space described in the previous section, where $\tau = 71.1$ns. At $t = 40$ns, the components of the spatial Fourier transform at the highest vertical and horizontal frequencies, denoted respectively as $(u, v) = (M/2, 0)$ and $(u, v) = (0, N/2)$, suffer a decrease on their magnitude by a factor of 0.1050 —substituting the values in Eq. (2), where the block in question is the complete image, thus $W = M$ and $H = N$. This means that their energy is reduced to just a 1.10% of its value at $t = 0$, so they have lost nearly 99% of their energy. It means that a subsampling factor equal to 2 can be applied over the vertical and horizontal dimensions of the image without significant lost of information. For $t = 80$ns (not shown in Fig. 8), both components $(u, v) = (M/2, 0)$ and $(u, v) = (0, N/2)$ have been even more attenuated and, additionally, $(u, v) = (M/4, 0)$ and $(u, v) = (0, N/4)$ have also lost around the

99% of their energy. In this case, a subsampling factor equal to 4 can be applied without losing relevant information. The resulting pyramids for two scale spaces generated on-chip are depicted in Fig. 9. Subsampling is realized during readout by making use of the capabilities for random access to the pixels' value implemented in the chip.

D. Multiresolution scene representations

The reconfigurability of the array together with the possibility of carrying out a complete diffusion, i. e. charge redistribution, within each block render the representation of a scene at different resolutions extremely flexible. Several examples directly extracted from the chip can be seen in Fig. 10. All the images but the last one correspond to different versions of homogeneous pixel binning. The last image represents a progressive coarse-to-fine division of the focal plane in order to achieve foveation of the scene. All these scene representations, are available immediately after photointegration. Apart from the exposure time, no extra time and no extra power are required to obtain them if the focal plane subdivision is already set.

E. Energy-based scene representations

This primitive has been satisfactorily tested by segmenting salient regions. The results are depicted in Fig. 11. In these scenes, the focal plane was divided into blocks of 8×8px. The total energy without any filtering, $V_{E_{ij}}$, and the remaining energy after a complete diffusion (t long enough), $V_{E_{ij,DC}} = V_{E_{ij}(t\to\infty)}$, were computed within every block. Thanks to the parallelism in the processing implemented by the array, the first computation took around 225ns while the second one, including the time interval of diffusion, around 1.2us. Once $V_{E_{ij}}$ and $V_{E_{ij,DC}}$ were extracted from the chip, $V_{E_{ij}} - V_{E_{ij,DC}}$ was calculated off-line for each block and normalized to its maximum value across the image. The same computations were ideally performed with MATLAB® over the original image. The accuracy of the chip for this operation is noticeable

inferior than for the scale space generation. The RMSE for the first example is 8.5% whereas for the second one is 10.9%, with respect to the ideal processing. The main source of error is the signal compression taking place at the generation of the energy representation. We have started with an image represented by the pixel voltages, V_{ij}. Each voltage is converted to a current by M_E according to the square-law of the MOS transistor. Therefore, any inaccuracy in the generation of V_{ij} is magnified by the square-law of the transistor. Right after that the current is linearly converted to voltage by discharging capacitor C_E. As the signal ranges for $V_{E_{ij}}$ and V_{ij} are similar, the signal representing the image energy is compressed compared to the signal representing the pixels' magnitude. Also second order effects, charge-injection errors, channel length modulation, transconductance and threshold mismatch, etc., become significant when millivolt range changes are usual. In any case, the absolute value of each block is not important in this case. The target of this processing is to segment the zones of the image with the largest changes of intensity, that is, the relative values among the blocks of the scene representation are the key point here. As can be seen in Fig. 11, the computation of the energy performed on-chip is capable of segmenting such zones. A subsequent step for a vision algorithm could be to realize dynamic foveation around the blocks with the largest values for a finer analysis. The outcome is depicted in Fig. 12 for the second scene of Fig. 11 Note that these foveated images, unlike that one in Fig. 10, keep full-resolution in the ROI but the minimum resolution possible, according to the programmability of the chip, in the rest of the scene.

To finish this section, Table II summarizes the power consumption for the different combinations of focal-plane processing, conversion and image size. All the figures are given at 30fps, although these frames are of a reduced size, as indicated in the first column that reflects the size in pixels of the blocks delivered. Keep in mind that the chip is not intended to deliver full frame images, but reduced representations of a high informational value. The measured power include the consumption of the A/D converter and the column buffers: 1.2mW (specifications,

not measured) and 0.8mW (measured), respectively. As a projection of the power consumption for a full frame output we can take into account that the current ADC and column buffers are able to deliver 0.11MSa/s (Table I), for what they need, roughly, 2.0mW. If 176×144-pixel frames are to be delivered at a rate of 30fps, what means 0.76MSa/s, we will need 7 times more power, i. e. 14.0mW. Notice that the power required for focal-plane processing is the same, as it is realized full-frame in parallel. The last column of Table II accounts for this projection. It gives an idea of the efficiency of the focal-plane processing proposed.

F. Comparative analysis

Several reported smart image sensors intend to efficiently implement image filtering and multiresolution representation. The performance indexes chosen to establish a comparison are area and power consumption, together with image resolution and throughput. Minimizing area and power consumption has been the driving force for the design of the FLIP-Q prototype. Regarding the accuracy of the processing, no comparison can be made in general. In most of the cases the operation of the reported image sensors is accurate enough for the corresponding target application but a thorough quantification of such accuracy is never given.

In [17], Gaussian filtering with user-defined σ is performed by means of a resistive network containing both positive and negative resistors. A very large power consumption is reported due mainly to the bias currents in the control circuit for the variable resistor. A simpler and more efficient implementation of this filtering is carried out in [18]. In this case, a solver of the spatially-discretized diffusion process is implemented by means of a capacitive network. The variance of the filter is determined by a capacitor ratio, fixed by layout design, and a iteration number associated to the implicit time discretization of the network. The main argument given in favour of this implementation instead of another one based on a dynamic RC network is that usually the time constant of the lattest is so small that sampling becomes difficult [19]. However,

we have demonstrated with the FLIP-Q prototype that this problem can be overcome by a fine on-chip control. Better accuracy is thus achieved in spite of the intrinsic nonlinearities of the transistors while performing not discrete but continuous-time diffusion. Regarding the area and power consumption associated to the specific operation of Gaussian filtering, no data is given in [18] to be compared with the performance of our prototype.

Vision chips capable of delivering programmable multiresolution scene representations have been also previously reported. In [20], capacitive networks outside the array are used to merge the pixel values. The main limitation of this chip is that its functionality is reduced to this operation. Besides, the blocks of pixels in which the image is divided must be square. The power consumption is of the same order of magnitude than that of the FLIP-Q prototype. The comparison in terms of area is more difficult to establish as the operation in [20] is not performed in-pixel but during the readout process. The die sizes, equalizing their resolutions by extrapolation, are very similar. Other processing arrays, like [21] and [22], use the multiresolution feature as a means to achieve a certain targeted outcome and therefore it is not separately characterized. In [21], the maximum possible reduction of resolution is by a factor of four outside the ROI while edge filtering at full-, half- and quarter resolution can be achieved in [22].

Table III summarizes the main reported features of the chips above commented. Although the functionalities of the prototypes do not exactly match, we have tried to compute a figure of merit that contemplates the major features of the chips: $\text{FOM} = (\text{Area} \cdot \text{Power})/(\text{Spatial resolution} \cdot \text{Throughput})$. From these results, it can be seen that the FLIP-Q prototype, implementing image processing tasks which are useful for most of vision algorithms, presents very competitive figures, specially in terms of power consumption. Chips with lower FOM, [20] and [21], do not perform Gaussian filtering. Those which realize this type of filtering have similar [18] or worse FOM [17], [22]. No chip delivering energy-based scene representations has been included in Table III. To the best of our knowledge, this simplificacion of the scene at the focal plane had not been

previously reported. Examples of other approaches for estimation of salient regions can be found in [23] and [24].

V. Conclusions

This paper has thoroughly described a smart CMOS image sensor intended for low-power applications. The prototype can deliver different degrees of simplification of a scene which alleviate the processing load of subsequent digital processing stages. Large signal behaviour of the transistors is greatly exploited in order to implement a massively parallel analog focal-plane array based on the SIMD paradigm. Experimental results show the enormous potential of the sensor and the energy efficiency of its operation.

Acknowledgment

The authors wish to thank J. M. Repiso, M. A. Lagos and J. M. Mora for their help with the test and development boards.

References

[1] K. Govil, E. Chan, and H. Wasserman, "Comparing algorithms for dynamic speed-setting of a low-power CPU," in *I Int. Conf. on Mobile Computing and Networking*, San Diego, USA, 1995, pp. 13–25.

[2] C. Poynton, *Digital Video and HDTV: Algorithms and Interfaces*. Elsevier Science, 2007.

[3] J. Dubois, D. Ginhac, M. Paindavoine, and B. Heyrman, "A 10000 FPS CMOS sensor with massively parallel image processing," *IEEE J. Solid-State Circuits*, vol. 43, no. 3, pp. 706–717, 2008.

[4] J. Poikonen, M. Laiho, and A. Paasio, "MIPA4k: A 64x64 cell mixed-mode image processor array," in *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2009, pp. 1927–1930.

[5] L. Zhiqiang, M. Hoffman, N. Schemm, W. Leon-Salas, and S. Balkir, "A CMOS image sensor for multi-level focal plane image decomposition," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 9, pp. 2561–2572, 2008.

[6] A. Nilchi, J. Aziz, and R. Genov, "Focal-plane algorithmically-multiplying CMOS computational image sensor," *IEEE J. Solid-State Circuits*, vol. 44, no. 6, pp. 1829 –1839, 2009.

[7] S. Unger, "A computer oriented toward spatial problems," *Proceedings of the IRE*, vol. 46, no. 10, pp. 1744–1750, 1958.

[8] T. Komuro, A. Iwashita, and M. Ishikawa, "A QVGA-size pixel-parallel image processor for 1000-FPS vision," *IEEE Micro*, vol. 29, no. 6, pp. 58–67, 2009.

[9] C. Chih-Chi, L. Chia-Hua, L. Chung-Te, and C. Liang-Gee, "iVisual: An intelligent visual sensor SoC with 2790 FPS CMOS image sensor and 205 GOPS/W vision processor," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 127–135, 2009.

[10] T. He, S. Krishnamurthy, J. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh, "Energy-efficient surveillance system using wireless sensor networks," in *MobiSys'04: Proceedings of the 2nd Int. Conf. on mobile systems, applications, and services*, 2004, pp. 270–283.

[11] M. Magno, D. Brunelli, L. Thiele, and L. Benini, "Adaptive power control for solar harvesting multimodal wireless smart camera," in *3rd ACM/IEEE Int. Conf. on Distributed Smart Cameras (ICDSC)*, 2009, pp. 1–7.

[12] T. Lindeberg, "Discrete scale-space theory and the scale-space primal sketch," Ph.D. dissertation, Royal Institute of Technology, Stockholm, Sweden, 1991.

[13] B. Jahne, H. Hauβecker, and P. Geiβler, *Handbook of Computer Vision and Applications*. Academic Press, 1999, vol. 2, ch. 4.

[14] J. Fernández-Berni and R. Carmona-Galán, "Accurate design of a MOS-based resistive network for time-controlled diffusion filtering," in *ECCTD*, Antalya, Turkey, 2009, pp. 683–686.

[15] Y. Ni, "Smart image sensing in CMOS technology," *IEE Proc.-Circuits Devices Syst.*, vol. 152, no. 5, pp. 547–555, 2005.

[16] J. Fernández-Berni and R. Carmona-Galán, "Robust focal-plane analog processing hardware for dynamic texture segmentation," in *12th Int. W. Cellular Nanoscale Networks and Apps. (CNNA)*, 2010, pp. 453–458.

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

[18] Y. Ni and J. Guan, "A 256x256 pixel smart CMOS image sensor for line-based stereo vision applications," *IEEE J. Solid-State Circuits*, vol. 35, no. 7, pp. 1055–1061, 2000.

[19] Y. Ni, Y.-M. Zhu, B. Arion, and F. Devos, "Yet another analog 2D gaussian convolver," in *IEEE International Symposium on Circuits and Systems*, vol. 1, 3-6 1993, pp. 192–195.

[20] S. Kemeny, R. Panicacci, B. Pain, L. Matthies, and E. Fossum, "Multiresolution image sensor," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 4, pp. 575–583, 1997.

[21] C. Jaehyuk, H. Sang-Wook, K. Seong-Jin, C. Sun-Il, and Y. Euisik, "A spatial-temporal multiresolution CMOS image sensor with adaptive frame rates for tracking the moving objects in region-of-interest and suppressing motion blur," *IEEE J. Solid-State Circuits*, vol. 42, no. 12, pp. 2978–2989, 2007.

[22] N. Takahashi, K. Fujita, and T. Shibata, "A pixel-parallel self-similitude processing for multiple-resolution edge-filtering analog image sensors," *IEEE Trans. Circuits Syst. I*, vol. 562, no. 11, pp. 2384–2392, 2009.

[23] V. Brajovic and T. Kanade, "Computational sensor for visual tracking with attention," *IEEE J. Solid-State Circuits*, vol. 33,

no. 8, pp. 1199–1207, 1998.

[24] K. Kwanho, L. Seungjin, K. Joo-Young, K. Minsu, and Y. Hoi-Jun, “A 125 GOPS 583 mW network-on-chip based parallel processor with bio-inspired visual attention engine,” *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 136–147, 2009.

Technology	0.35μm CMOS 2P4M
Vendor (Process)	Austria Microsystems (C35OPTO)
Die size (with pads)	7280.8μm $\times$ 5780.8μm
Cell size	34.07μm $\times$ 29.13μm
Fill factor	6.45%
Resolution	QCIF: 176$\times$144 px
Photodiode type	n-well/p-substrate
Power supply	3.3V
Signal range	[1.5V,2.5V]
FPN	0.72%
PRNU (50% signal range)	2.42%
Sensitivity	0.15V/(lux·s)
Measured power consumption	5.6mW@30fps
(worst case)	22 $\times$ 18px
Predicted power consumption	17.6mW@30fps
(worst case)	176 $\times$ 144px
ADC throughput	0.11MSa/s (9μs/Sa)
Internal clock freq. range	0.5-150MHz

TABLE I

SUMMARY OF THE PROTOTYPE CHIP FEATURES.

Block (px)	**VCO freq. (MHz)**	**Diffusion steps (N_1)**	**Energy comput.**	**Power consumption (mW)**	**Predicted power for full-frame output (mW)**
4×4	VCO off	External diffusion control	No	2.7	14.7
4×4	5	5	No	2.9	14.9
4×4	50	5	No	3.5	15.5
4×4	150	10	No	5.4	17.4
8×8	150	10	Yes	5.6	17.6
8×8	VCO off	No diffusion	Yes	2.9	14.9
8×8	VCO off	No diffusion	No	2.0	14.0

TABLE II

POWER CONSUMPTION OF THE CHIP FOR DIFFERENT FOCAL-PLANE PROCESSING CONFIGURATIONS.

Author / Reference	**Tech. (μm) / Year**	**Processing capabilities**	**Die size (mm^2)**	**Array size**	**Cell size (μm^2)**	**Power (mW)**	**Throughput (MSa/s)**	**FOM (pJ·mm^2/px·Sa)**
Kobayashi [17]	2 / 1991	Gaussian filtering	7.9 × 9.2	45×40	170 × 200	2000	0.054	1.49×10^6
Kemeny [20]	1.2 / 1997	Multiresolution imaging	4.8 × 6.6	128 × 128	24 × 24	5	0.49	19.7
Ni [18]	0.8 / 2000	Analog histogram equalizer, Gaussian and DoG filtering and local extrema extractor	7 × 7	256 × 256	20 × 20	200 (worst case)	1.57	95.1
Choi [21]	0.35 / 2007	Multiresolution imaging with ROI estimation from motion detection	5 × 5	256 × 256	8.9 × 8.9	74.87	1.97	14.5
Takahashi [22]	0.35 / 2009	Edge filtering	9.8 × 9.8	64 × 64	123.3 × 124.8	350	2.79	2.95×10^3
This work	0.35 / 2010	Scale space and pyramid generation, multiresolution imaging and energy-based representation	7.28 × 5.78	176×144	34.07 × 29.13	5.6 (worst case)	0.11	84.5

TABLE III

COMPARISON OF FOCAL-PLANE PROCESSING CHIP PERFORMANCE

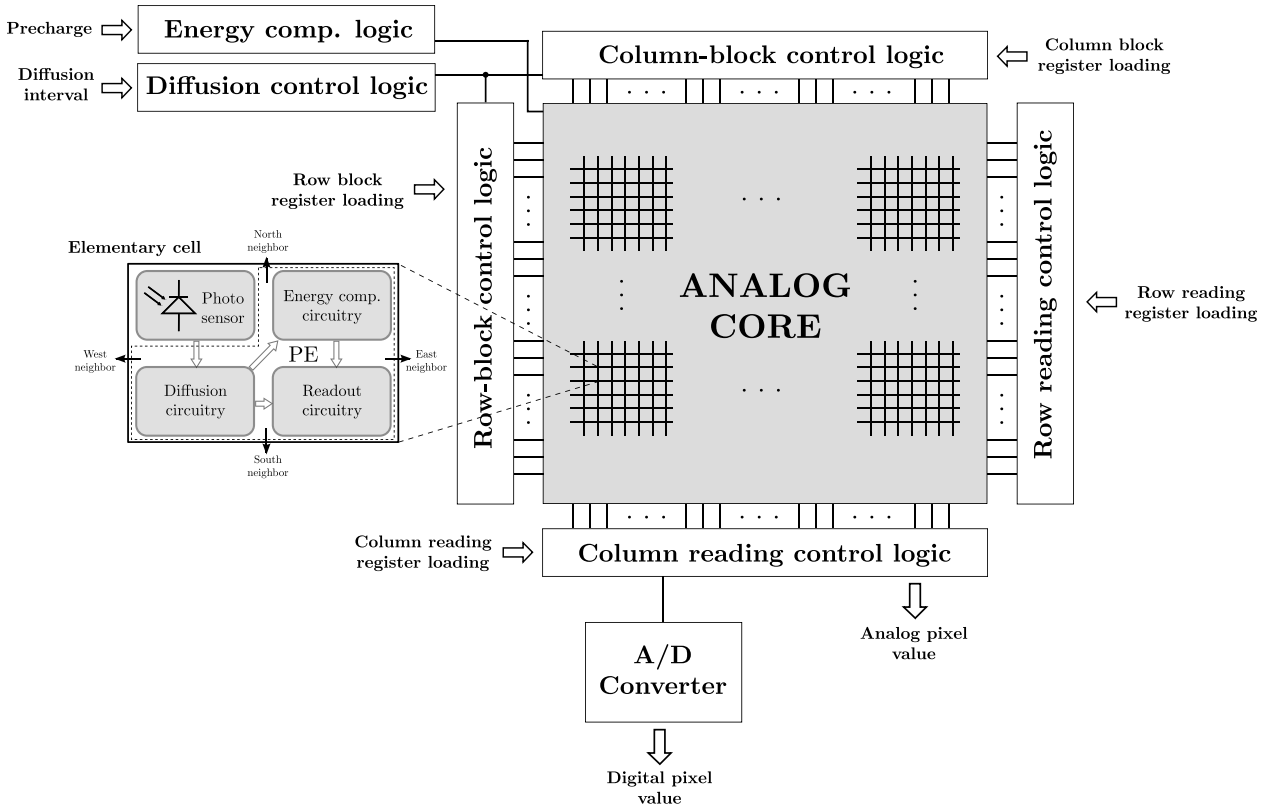


Fig. 1. Floorplan of the prototype chip.

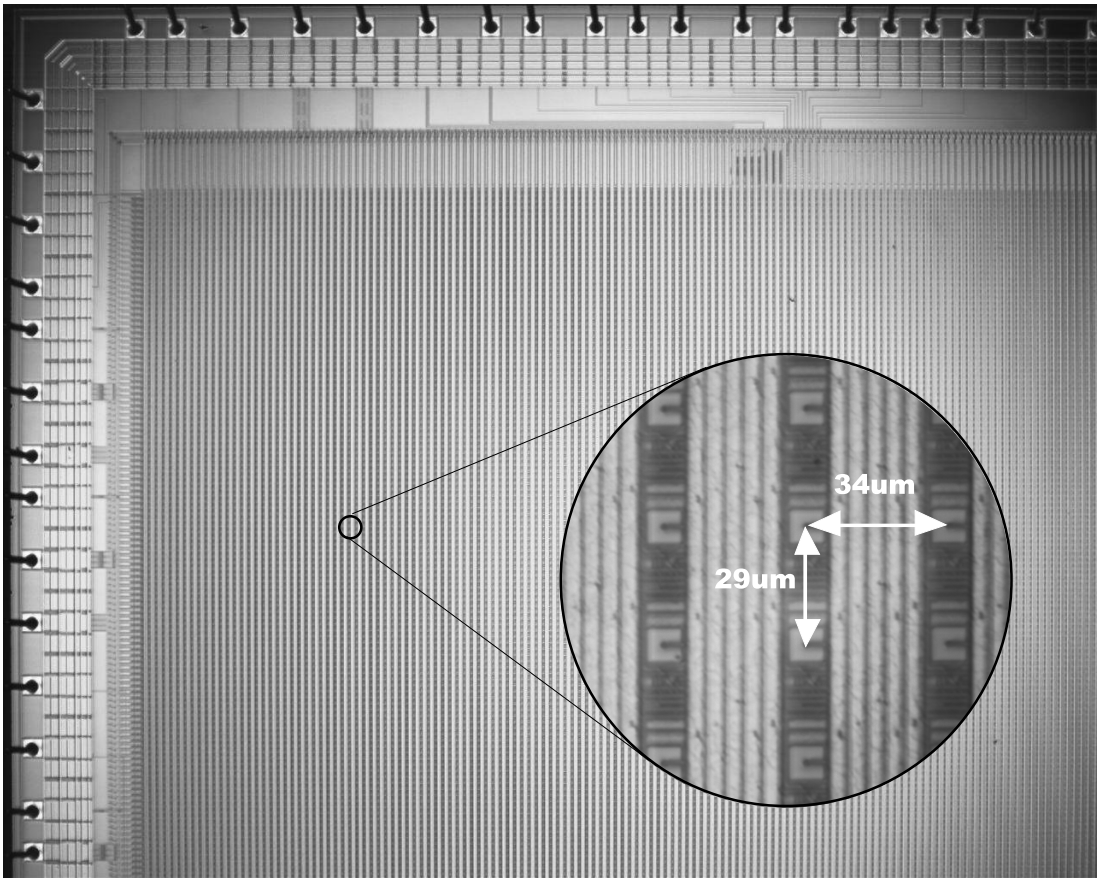


Fig. 2. Microphotographs of the FLIP-Q prototype chip.

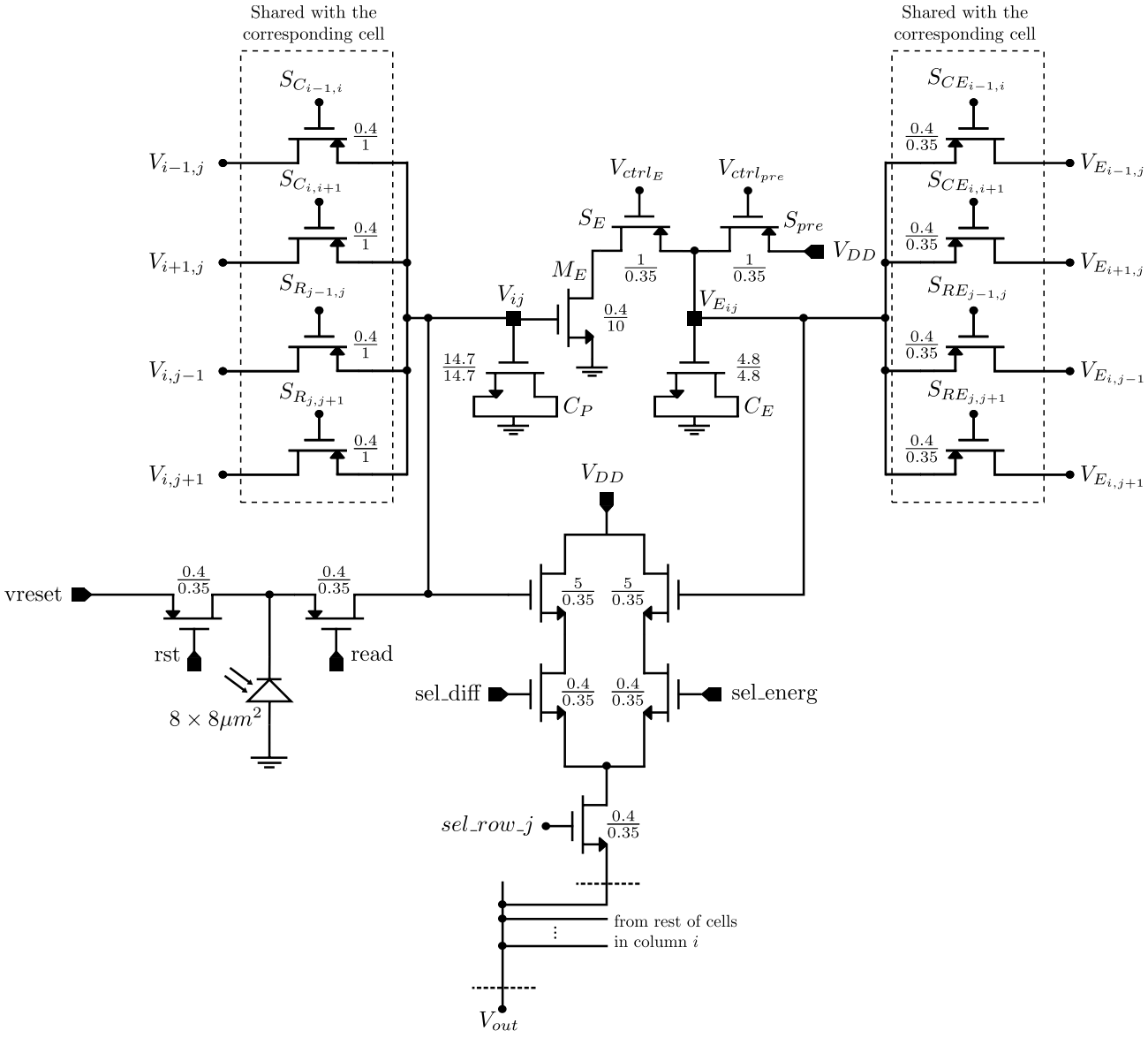


Fig. 3. Elementary cell of the array.

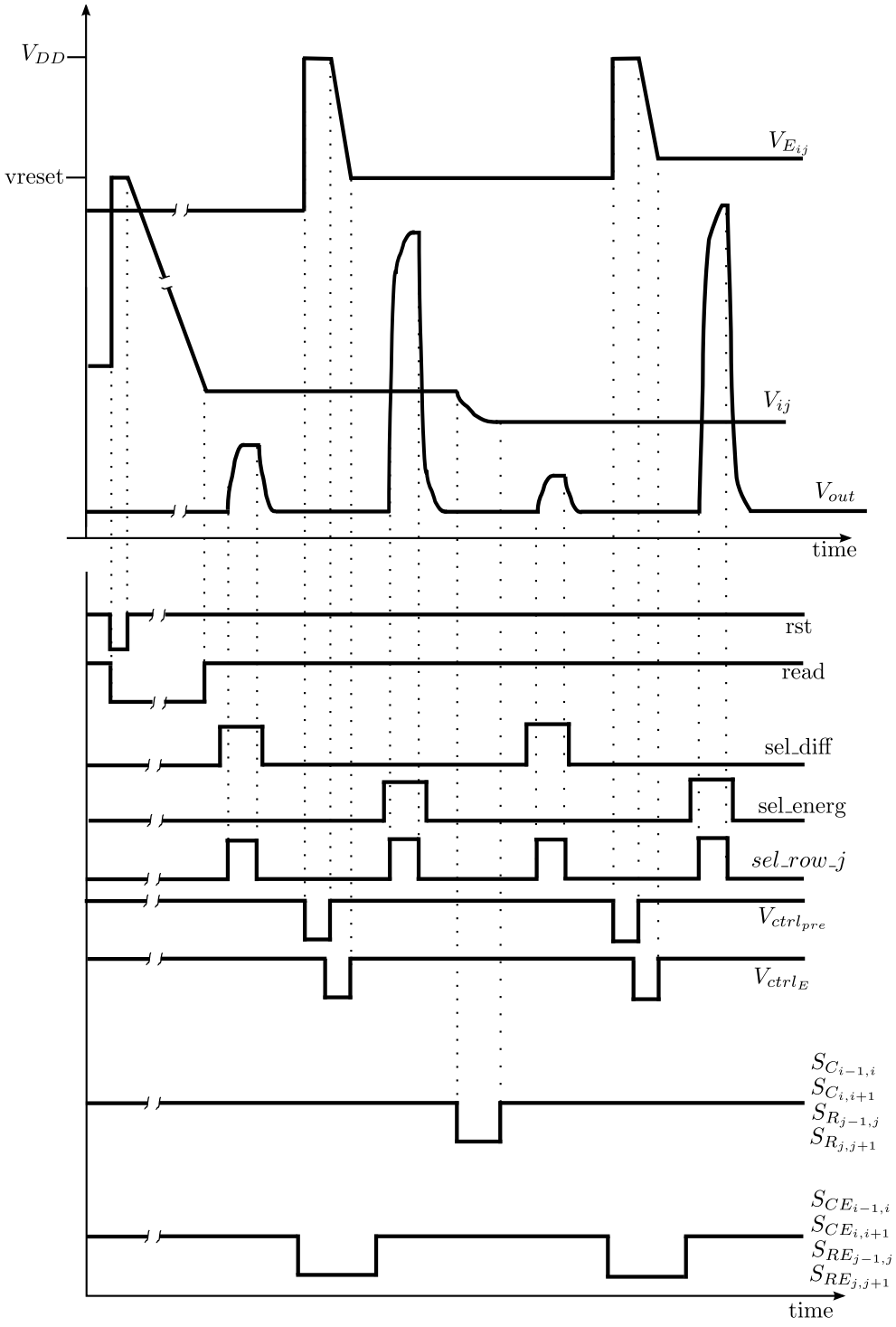


Fig. 4. Timing diagram of the operation of the elementary cell.

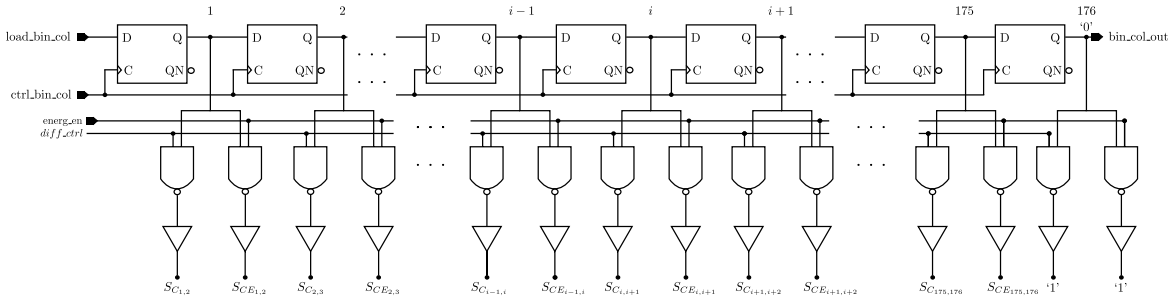


Fig. 5. Column-wise focal-plane division control.

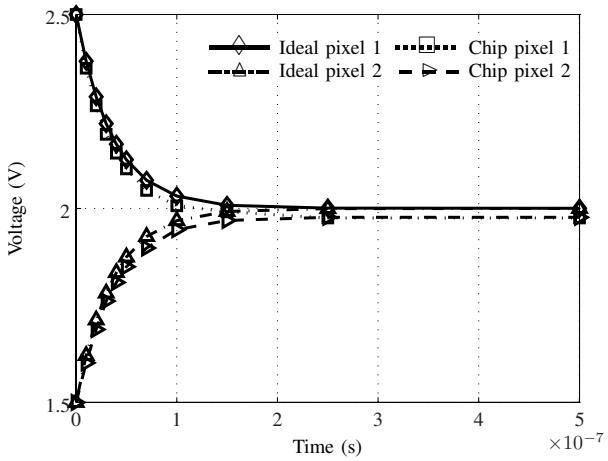


Fig. 6. Calibration of τ at the upper left corner

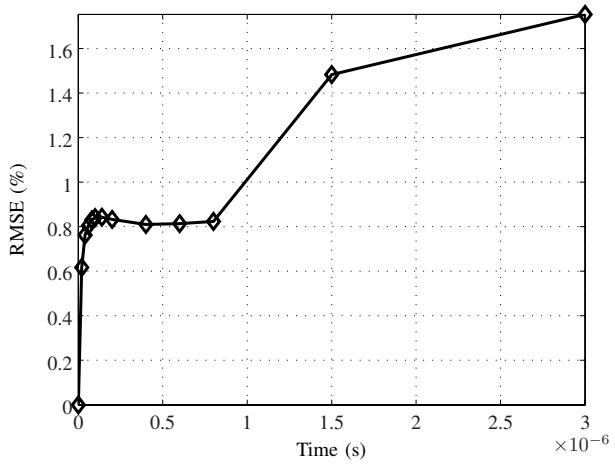


Fig. 7. RMSE for the on-chip scale space with respect to the ideal case

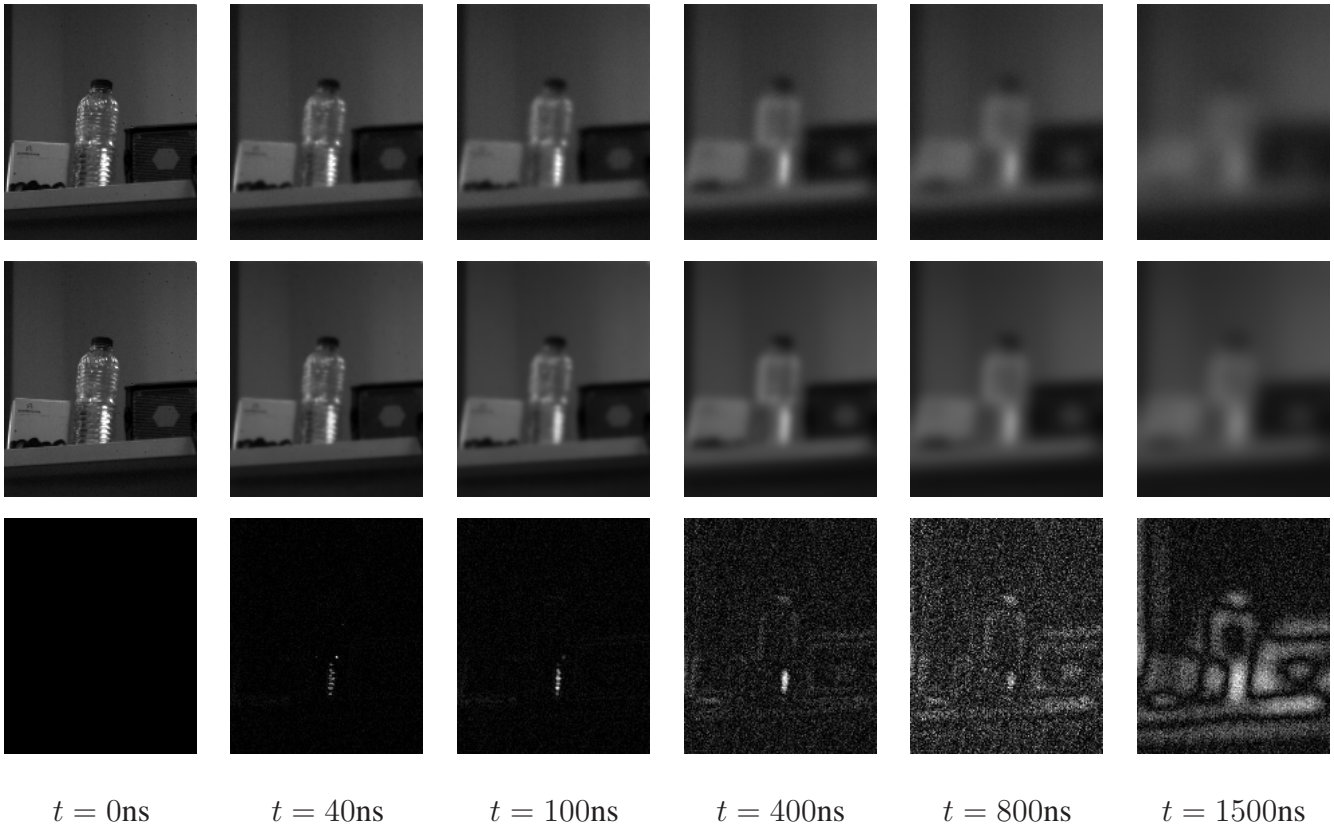


Fig. 8. Scale spaces along time. The first row corresponds to the on-chip scale space, the second one corresponds to the ideal scale space and finally the third one corresponds to their normalized difference.

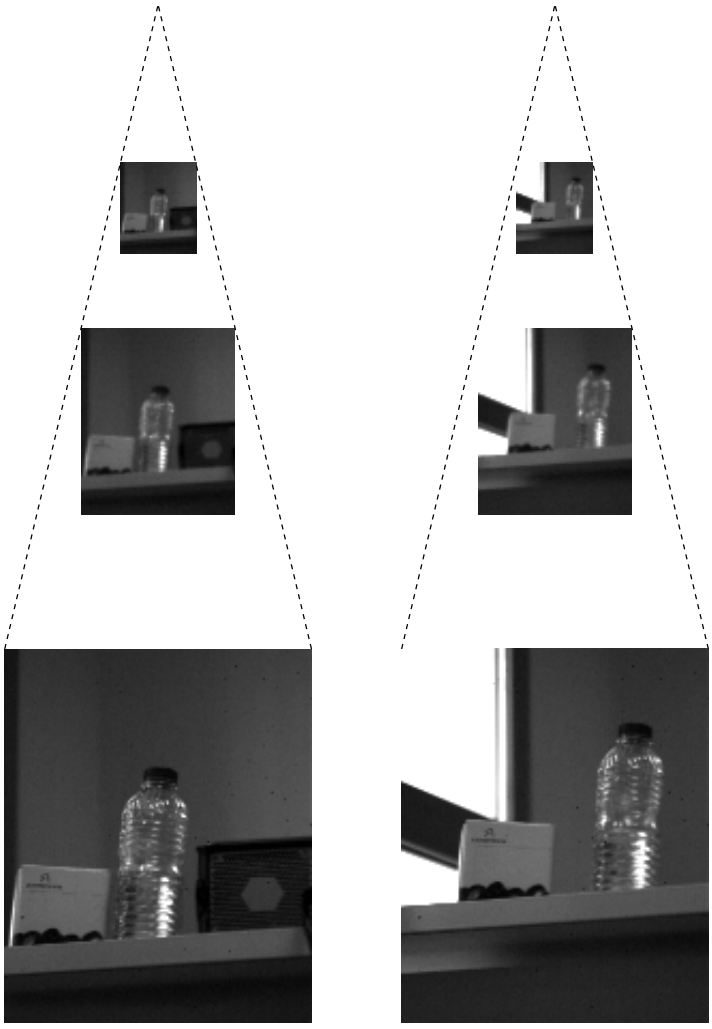
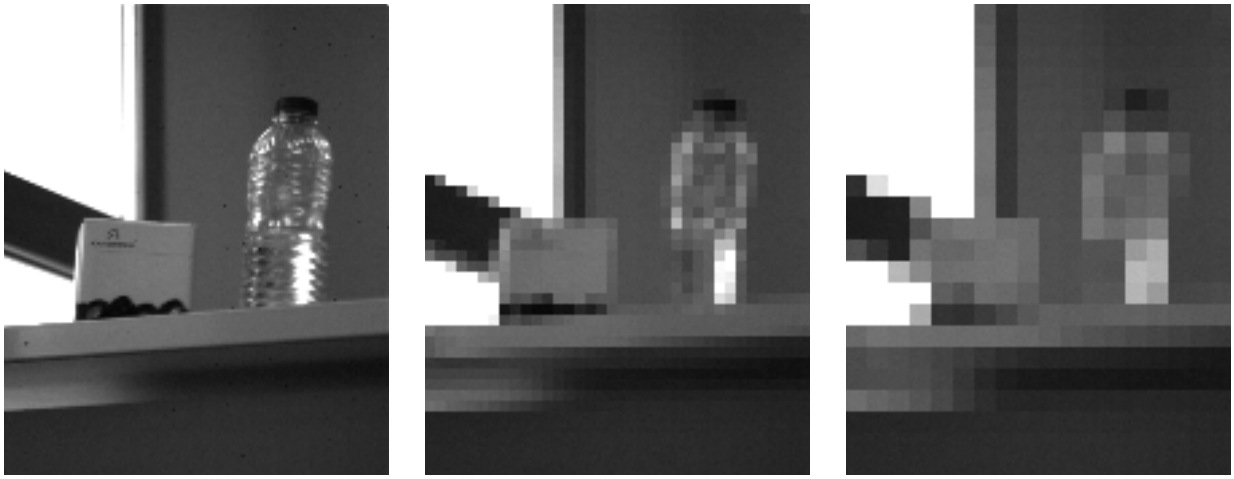
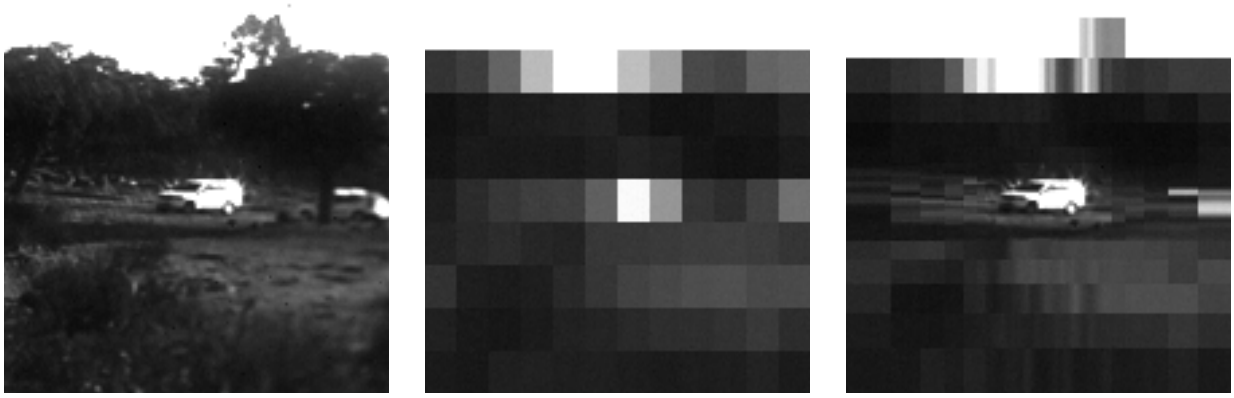


Fig. 9. Pyramid representation of two on-chip scale spaces.

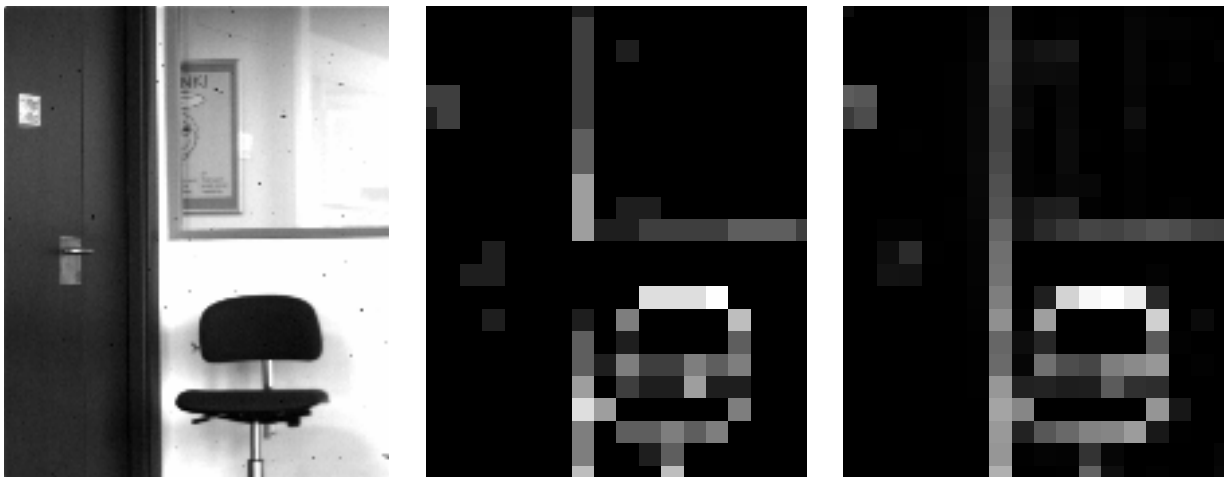


Original image 4×4 px 8×8 px

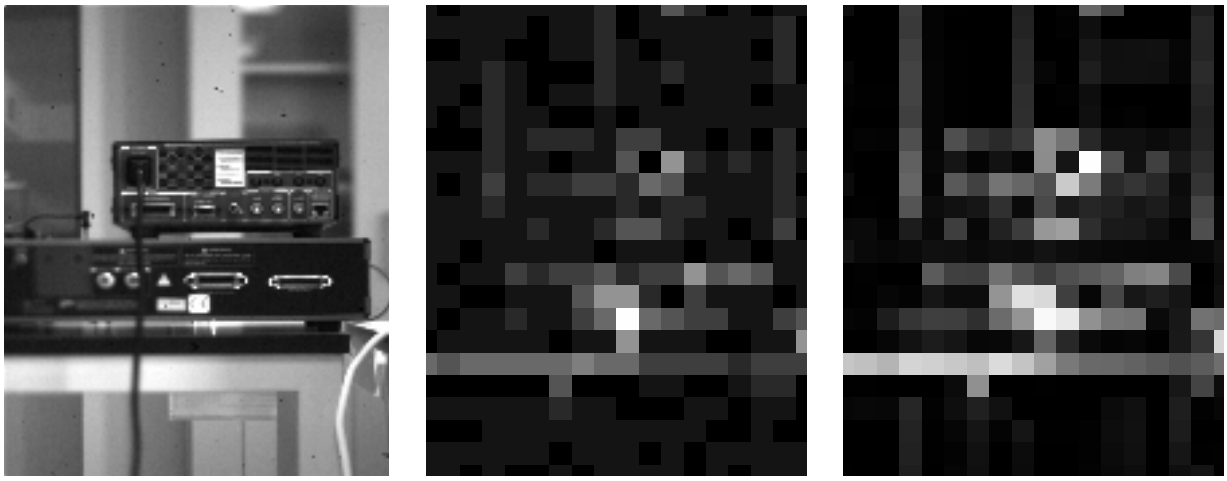


Original image 12×16 px Foveation

Fig. 10. Examples of multiresolution scene representation



Original image | On-chip processing | Ideal processing



Original image | On-chip processing | Ideal processing

Fig. 11. Examples of energy-based scene representation

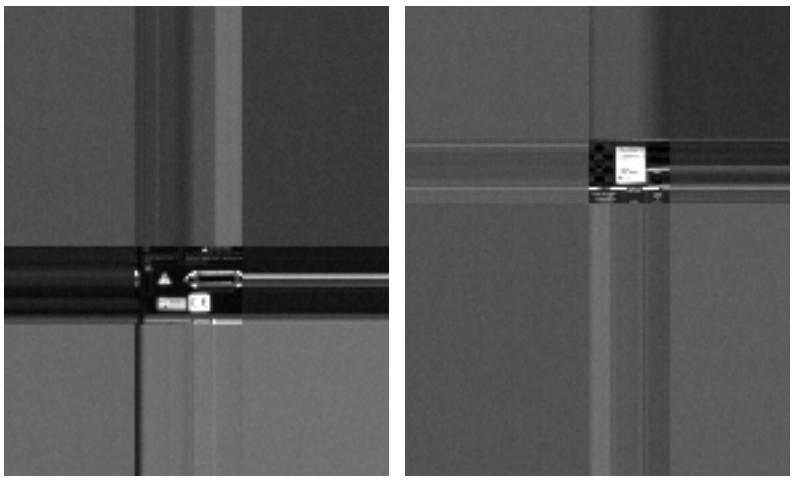


Fig. 12. On-chip abrupt foveation around the blocks segmented by the computation of the energy in the second scene of Fig. 11.