Smart imaging for power-efficient extraction of Viola-Jones local descriptors

J. Fernández-Berni[a], R. Carmona-Galán[a], R. del Río[a], Juan A. Leñero-Bardallo[a],
M. Suárez-Cambre[b] and Á. Rodríguez-Vázquez[a]

[a]Institute of Microelectronics of Seville (IMSE-CNM), CSIC-Universidad de Sevilla, Spain
[b]Centro de Investigación en Tecnologías de la Información (CITIUS), University of Santiago de Compostela, Spain

ABSTRACT

In computer vision, local descriptors permit to summarize relevant visual cues through feature vectors. These vectors constitute inputs for trained classifiers which in turn enable different high-level vision tasks. While local descriptors certainly alleviate the computation load of subsequent processing stages by preventing them from handling raw images, they still have to deal with individual pixels. Feature vector extraction can thus become a major limitation for conventional embedded vision hardware. In this paper, we present a power-efficient sensing-processing array conceived to provide the computation of integral images at different scales. These images are intermediate representations that speed up feature extraction. In particular, the mixed-signal array operation is tailored for extraction of Haar-like features. These features feed the cascade of classifiers at the core of the Viola-Jones framework. The processing lattice has been designed for the standard UMC 0.18µm 1P6M CMOS process. In addition to integral image computation, the array can be reprogrammed to deliver other early vision tasks: concurrent rectangular area sum, block-wise HDR imaging, Gaussian pyramids and image pre-warping for subsequent reduced kernel filtering.

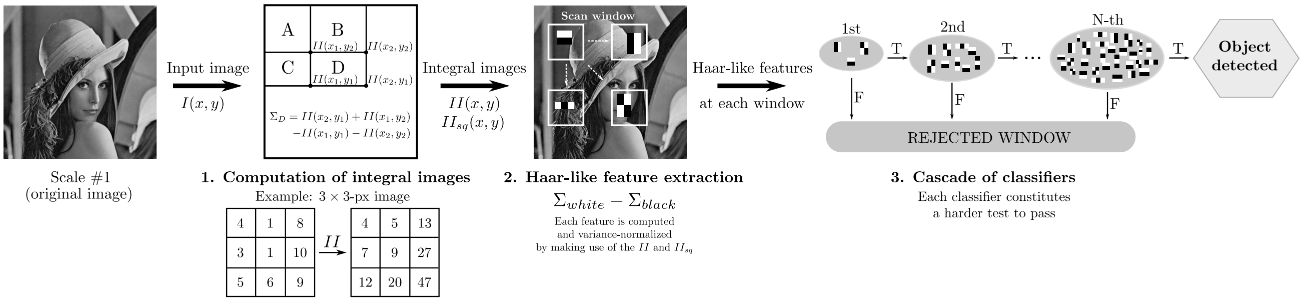
Keywords: Viola-Jones algorithm, smart imaging, sensing-processing arrays, mixed-signal circuitry, Haar-like features, OpenCV library, integral images.

1. INTRODUCTION

Feature detectors are widely used for computer vision applications such as object detection and classification, image retrieval, 3-D reconstruction or tracking, among others.[1] They are based on the extraction of local descriptors at *early vision* stages encoding relevant visual cues conveyed by means of feature vectors. These vectors constitute inputs for trained classifiers or matching algorithms which in turn enable different high-level vision tasks. While local descriptors certainly alleviate the computation load of subsequent processing stages by preventing them from handling raw images, they still have to deal with individual pixels. Feature vector extraction can thus become a major limitation for conventional embedded vision hardware.

Focal-plane sensing-processing[2] constitutes the best approach in terms of exploitation and adaptation to the particular characteristics of early vision.[3] On the one hand, the information to be handled at this processing stage —each and every pixel resulting from the raw readings of the sensors— is massive. On the other hand, the computational flow is very uniform. The same calculations are repeatedly carried out on every pixel. More interestingly, the outcome for each individual pixel does not usually depend on the outcome for the rest. Consequently, while an enormous amount of data must certainly be processed, regular massively parallel operation can still be applied. Focal-plane sensor-processor chips make the most of these characteristics by operating in Single Instruction Multiple Data (SIMD) mode[4] featuring concurrent processing and distributed memory. Focal-plane processing architectures can also benefit from the possibility of including *analog circuitry*. When compared to their digital counterpart, analog circuits can reach higher performance in terms of speed, area and power consumption, but at the cost of low, moderate at most, accuracy. Fortunately, most vision algorithms can perform

Further author information:
Jorge Fernández-Berni: C/ Américo Vespucio s/n, 41092, berni@imse-cnm.csic.es, telephone: +34 954 46 66 66



4	1	8
3	1	10
5	6	9

4	5	13
7	9	27
12	20	47

Figure 1. Simplified scheme of the Viola-Jones processing flow. It is applied over successive scaled versions of the original image.

properly under these conditions.[5] Numerous smart image sensors have been successfully implemented following this scheme.[6–10] Even commercial general-purpose vision systems based on focal-plane processing have been reported.[11]

All in all, we address in this paper the design of focal-plane mixed-signal array circuitry implementing local descriptors required by the Viola-Jones processing framework.[12] In addition to other capabilities, the resulting array provides support to the Viola-Jones processing flow at two low-level stages. First, it is be able to deliver the integral and square integral images at different scales. Alternatively, it can compute the sum of pixels and squared pixels at any possible rectangular area of the image, significantly easing the extraction of Haar-like features. And this is achieved while making the most of two inherent characteristics of focal-plane operation: distributed memory and ultra-low-power consumption.

2. VIOLA-JONES PROCESSING FRAMEWORK

The Viola-Jones framework constitutes one of the best approaches reported to achieve real-time object recognition. It is based on the extraction of very simple features across the image —the so-called *Haar-like features*— which are subsequently analyzed by a cascade of classifiers of progressive complexity. These classifiers are previously trained according to the object to be detected, adapting their internal thresholds when successive training images are passed through. A basic scheme of the Viola-Jones processing flow is depicted in Fig. 1. Despite its simplicity, this framework still requires a considerable amount of computational and memory resources. During the last few years, numerous efforts have been focused on exploiting the increasing memory and logic capabilities available in FPGAs[13] as well as the highly parallel computation structure of GPUs.[14] When it comes to low-power embedded systems, additional constraints must be introduced on the image resolution[15] or the type of processor operations[16] in order to obtain at least moderate frame rates.

From the point of view of focal-plane processing, we are interested in the early stages of the flow represented in Fig. 1. From now on, we will be considering the most usual operation mode for the Viola-Jones framework.[16] In this mode, the original image is scaled until reaching a prescribed minimal dimension. The processing flow is repeated for each scaled image. Note that the raw material feeding the cascade of classifiers consists of Haar-like features. These features derive from the Haar wavelets[17] and encode differences in average intensities between rectangular regions. Their mathematical formulation is extremely simple. For a certain feature F_k, we have that:

$$F_k = \sum_i \sum_j W_{ij} - \sum_m \sum_n B_{mn} \tag{1}$$

where W_{ij} represents the pixel values within the white rectangle/s and B_{mn} the pixel values for the black rectangle/s. White and black are mere indicators of the area considered, with independence of their pixel values. In practice, we are simply comparing the DC component of the rectangles involved since the sum of the pixels is proportional to their mean value.

The large amount of resources to be allocated for the algorithm comes from the correspondingly large number of Haar-like features to compute. As an example, the Viola-Jones face detection algorithm provided by the

OpenCV library[18] requires 22 classifiers including 2135 features in total. Of course, most of the windows scanned across the image are rejected at the first —and simpler— classifiers of the cascade on not containing the targeted object. This avoids a great deal of useless calculations. But still there will be windows in which all the features will have to be checked. In order to alleviate the computational and memory requirements from this processing stage, an intermediate image representation is used, the so-called *integral image*. This intermediate representation is defined as:

$$II(x,y)=\sum_{x'=1}^{x}\sum_{y'=1}^{y} I(x',y') \tag{2}$$

where $I(x,y)$ represents the input image. That is, each pixel composing $II(x,y)$ is given by the sum of all the pixels above and to the left of the corresponding pixel at the input image. Two fundamental advantages support the inclusion of this pre-processing stage. First of all, only four pixels adequately extracted from the integral image permit to compute the sum of any rectangular region of the input image. Consider four points as in Fig. 1, $(x_1,y_1),(x_2,y_1),(x_1,y_2)$ and (x_2,y_2), with $x_1 < x_2$ and $y_1 < y_2$, defining a rectangle across the input image. The sum of pixels within this region can be expressed as:

$$\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} I(x,y)=II(x_2,y_1)+II(x_1,y_2)-II(x_1,y_1)-II(x_2,y_2) \tag{3}$$

The second advantage is that the integral image can be computed in one pass over the input image by making use of the following pair of recurrences:

$$\begin{cases} r(x,y)=r(x,y-1)+I(x,y) \\ II(x,y)=II(x-1,y)+r(x,y) \end{cases} \tag{4}$$

with $r(x,0)=0$ and $II(0,y)=0$. This single-pass computation enables a fast operation on the part of a microprocessor.

In addition to the integral image, the Viola-Jones processing flow also requires the calculation of the *square integral image*, defined as:

$$II_{sq}(x,y)=\sum_{x'=1}^{x}\sum_{y'=1}^{y} I^2(x',y') \tag{5}$$

This extra intermediate representation allows, in conjunction with $II(x,y)$, the variance normalization of the Haar-like features. All the windows used for classifier training are variance-normalized in order to minimize the effect of different lighting and contrast conditions. Correspondingly, the features extracted from the input image must also be variance-normalized. Taking into account that the variance of the generic rectangle previously defined for Eq. (3) can be expressed as:

$$\sigma^2=\frac{1}{WH}\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} I^2(x,y)-\left[\frac{1}{WH}\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} I(x,y)\right]^2 \tag{6}$$

and considering the counterpart of Eq. (3) for $II_{sq}(x,y)$, we can re-write Eq. (6) as:

$$\begin{aligned}\sigma^2 \;=\; & \tfrac{1}{WH}\left[II_{sq}(x_2,y_1)+II_{sq}(x_1,y_2)\;-II_{sq}(x_1,y_1)-II_{sq}(x_2,y_2)\right] \\ & -\left\{\tfrac{1}{WH}\left[II(x_2,y_1)+II(x_1,y_2)\;\;-II(x_1,y_1)-II(x_2,y_2)\right]\right\}^2\end{aligned} \tag{7}$$

which shows how both integral images work together to achieve the variance normalization.

In the next section, we will propose a processing scheme devised for the focal-plane computation of $II(x,y)$ and $II_{sq}(x,y)$. This computation, that constitutes the lowest-level task for the Viola-Jones framework, can

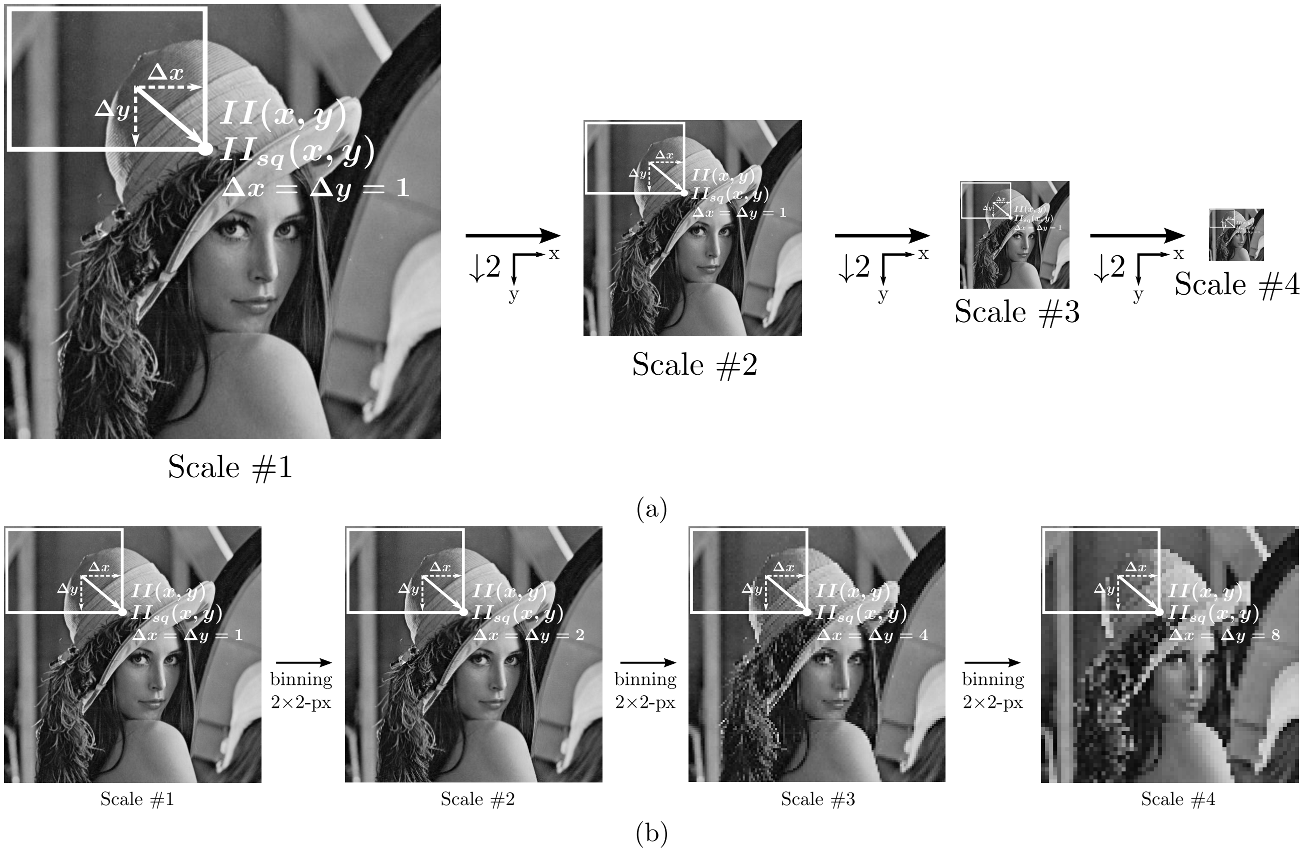


Figure 2. Integral images are to be computed from the original image and successively downsampled versions of it (a). In practice, this is equivalent to compute them from the original image and its successive versions obtained by pixel binning (b).

clearly benefit from the concurrent operation and distributed memory provided by focal-plane architectures. We will also demonstrate that the same scheme can directly deliver the sum of pixels and squared pixels at multiple rectangular image areas in parallel. This means that the computation of the integral images would not be really needed. However, the full exploitation of this characteristic falls beyond the scope of this paper. It would demand new algorithm-level strategies driving a Haar-like feature extraction tailored for the concurrent sum of rectangular regions across the image.

3. FOCAL-PLANE IMPLEMENTATION OF VIOLA-JONES EARLY VISION TASKS

Our objective is the implementation of reconfigurable focal-plane circuitry delivering integral images at different scales, as depicted in Fig. 2(a). For the sake of hardware simplicity, this is equivalent in practice to make use of the original image and its successive versions obtained by pixel binning, as represented in Fig. 2(b). For each scale, the pixels are correspondingly merged through averaging and the computation step along the x and y axis is doubled. In order to reach the aforementioned objective, we propose a general scheme like that of Fig. 3. A focal-plane array of 4-connected sensing-processing elementary cells provides the computational and memory resources required. These cells, whose interconnection can be reconfigured by means of peripheral circuitry, operate in a massively parallel way. They will work concurrently and jointly according to the corresponding instruction. Note however that such parallelism cannot be applied to obtain all the pixels of an integral image at the same time. Assuming a $W \times H$ array, it would mean to hold $W \times H$ copies of the top-left pixel of the original image since this pixel is needed for the computation of each and every pixel of the integral image. Likewise, a progressively reduced number of pixel copies along the original image would also have to be held. Instead,

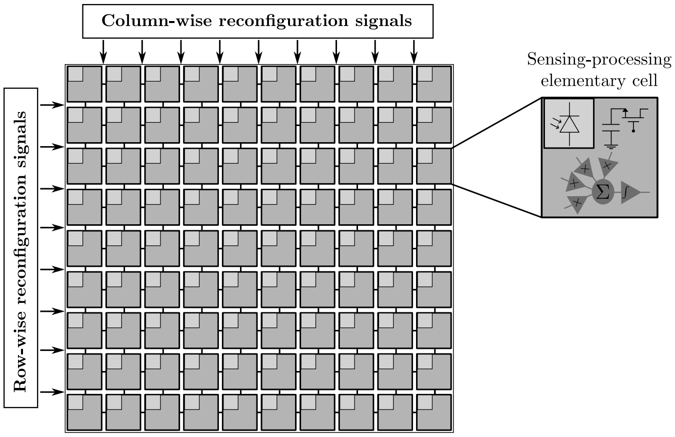

Column-wise reconfiguration signals
Row-wise reconfiguration signals
Sensing-processing
elementary cell
Σ
∫

Figure 3. Focal-plane sensing-processing scheme proposed for the computation of integral images.

we propose the exploitation of concurrency and distributed memory during a sequential processing stage. The circuitry to achieve this is shown in Fig. 4. It has been designed for the standard UMC 0.18μm 1P6M CMOS process. After photointegration, the pixel is represented by voltage V_{ij}. This voltage is repeatedly copied into $V_{S_{ij}}$ by enabling the analog buffer through the control signal CP_EN and then squared at $V_{SQ_{ij}}$ in order to respectively support the computation of the integral and square integral images. We re-use the squarer reported in[19] because of its simplicity and successful experimental verification. The sum of pixels and squared pixels is carried out through charge redistribution enabled by the switches controlled by $\mathrm{EN}_{S_{i,i+1}}$, $\mathrm{EN}_{S_{j,j+1}}$, $\overline{\mathrm{EN}_{SQ_{i,i+1}}}$ and $\overline{\mathrm{EN}_{SQ_{j,j+1}}}$. These signals are set by peripheral circuitry according to the scale and current pixel location of the integral images being calculated, as will be explained shortly. Once redistributions take place in parallel at $V_{S_{ij}}$ and $V_{SQ_{ij}}$, these voltages constitute new pixels of the targeted integral images. Charge redistribution can really be described as a diffusion process defined, for example for $V_{S_{ij}}$, as:

$$R_S C_S \frac{dV_{S_{ij}}}{dt} = -4V_{S_{ij}} + V_{S_{i+1,j}} + V_{S_{i-1,j}} + V_{S_{i,j+1}} + V_{S_{i,j-1}} \tag{8}$$

where R_S is the equivalent resistance of the switches and C_S is the capacitance holding $V_{S_{ij}}$. Eq. (8) is for an inside cell like that of Fig. 4 featuring full connectivity. Cells at the edges are connected to fewer than four neighbors. Unlike previous implementations,[19] we are not now interested in transient states of this diffusion process but in the steady state. And we want to attain it as fast as possible. Consequently, the switches are as wide as area restrictions allow, thereby reducing their resistance. The steady state of a diffusion process like that of Eq. (8) is characterized by a uniform distribution of voltages across the group of cells involved. Every voltage reaches the same value, that coincides with the average of the initial voltages at the cells. It is this average what encodes the sum required by the integral images.

In order to better visualize how the charge redistribution is configured, a simplified scheme of the proposed array is shown in Fig. 5. It can be seen that the cells can be grouped column-wise and row-wise through the corresponding control signals. Each pixel of the integral images is related to a stage of copy, squaring and charge redistribution. After these three steps, the array must be re-arranged for the next pixel. As an example, the computation of the first row of the integral images at scale #1 requires to disable all row connections between cells and then progressively enable column connections. Thus, if we focus on $\mathrm{EN}_{S_{i,i+1}}$, the column interconnection pattern '0000...0' leads to $II(1,1)$, '1000..0' to $II(2,1)$, '1100..0' to $II(3,1)$ and so on. Applying ones' complement to these patterns, those of $\overline{\mathrm{EN}_{SQ_{i,i+1}}}$ for $II_{sq}(1,1)$, $II_{sq}(2,1)$, $II_{sq}(3,1)$, etc. are respectively obtained. For further scales, a more complex redistribution arrangement is needed. To explain this, let us describe peripheral circuitry capable of providing $\mathrm{EN}_{S_{i,i+1}}$ and $\overline{\mathrm{EN}_{SQ_{i,i+1}}}$ —exactly the same is used for $\mathrm{EN}_{S_{j,j+1}}$ and $\overline{\mathrm{EN}_{SQ_{j,j+1}}}$. It is depicted in Fig. 6. We basically require a shift register.[19] This makes reconfiguration for scale #1 very simple and also ease further processing capabilities like image pre-warping for subsequent reduced kernel filtering.[20] But the point is how to deal efficiently with successive scales. Keep in mind that we first need pixel

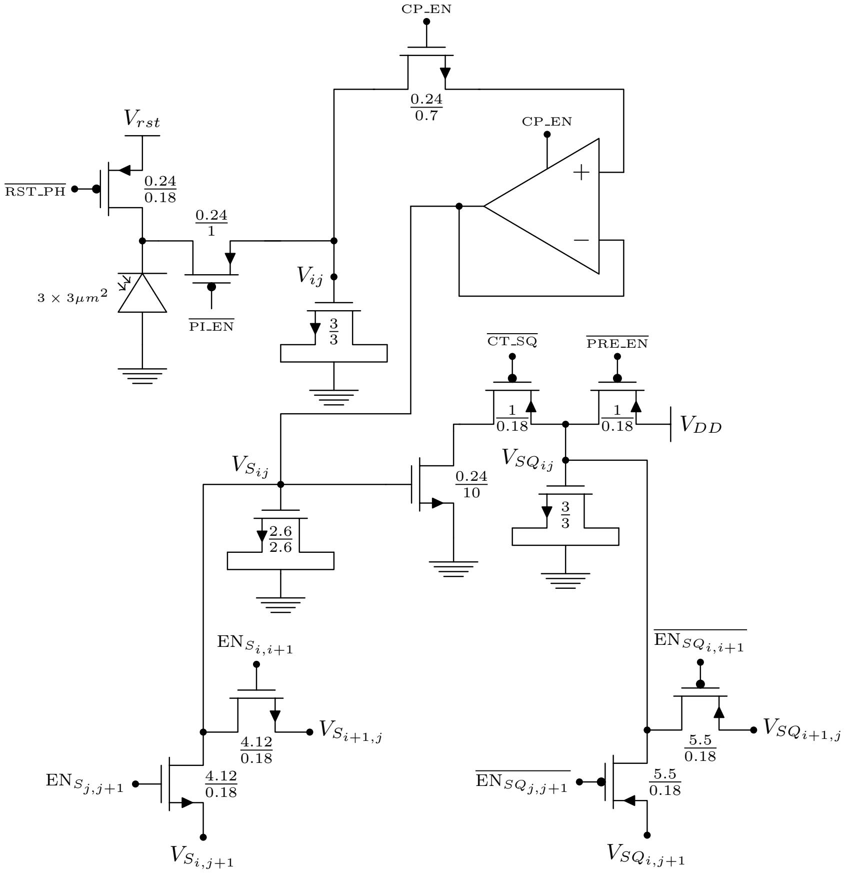


Figure 4. Proposed circuitry for integral image computation at the elementary sensing-processing cell.

binning and then reconfigurable charge redistribution over the resulting image. To speed up these two tasks, we incorporate the possibility of disabling the shift register and setting interconnection patterns in parallel through the signals denoted as $SC\#$. These signals are distributed along the peripheral cells as illustrated in Fig. 7. For scale #1, binning is not necessary. Our starting point is therefore an all-0's bit string for rows and columns. For scale #2, binning is achieved by switching the signal $SC2$, distributed as indicated in Fig. 7, from logic '0' to '1'. In so doing, we merge voltages $V_{S_{ij}}$ and $V_{SQ_{ij}}$ within 2×2-px blocks. By switching also the signal $SC3$ to '1', again as distributed in the figure, the merging process would affect blocks of 4×4-px. Likewise, $SC4$ is associated with 8×8-px blocks and $SC5$ with 16×16-px blocks. A single signal therefore permits to re-arrange the array for the next scale. The final step is to perform charge redistribution between the macro-pixels thus generated. This can be done by loading the adequate interconnection patterns, similarly to scale #1. Taking scale #2 as an example, and assuming again the computation of the first row, the column interconnection pattern '1000...0' would lead to $II(1,1)$, '111000000..0' to $II(2,1)$, '111110000..0' to $II(3,1)$, etc. These patterns mean to double the computation step along the x axis with respect to that of scale #1, accordingly to the macro-pixel dimensions. It is this enormous flexibility for focal-plane reconfiguration what endows the array with the additional asset of computing the sum of pixels and squared pixels at multiple rectangular areas in parallel, as required for the direct extraction of Haar-like features. The reconfigurability can also be exploited for block-wise intra-frame HDR imaging.[21]

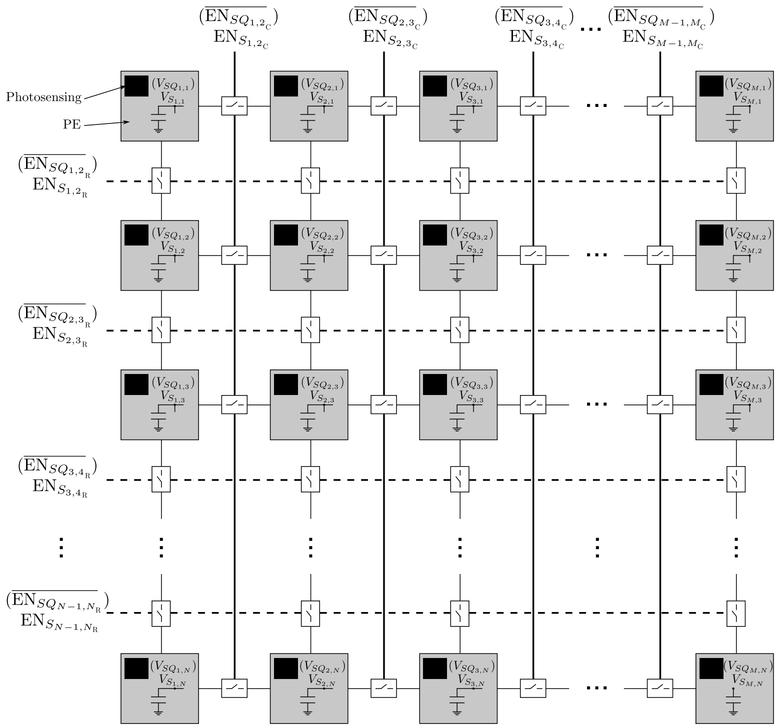

$(\overline{\mathrm{EN}_{SQ_{1,2_C}}})$ $\mathrm{EN}_{S_{1,2_C}}$
$(\overline{\mathrm{EN}_{SQ_{2,3_C}}})$ $\mathrm{EN}_{S_{2,3_C}}$
$(\overline{\mathrm{EN}_{SQ_{3,4_C}}})$ $\mathrm{EN}_{S_{3,4_C}}$
$(\overline{\mathrm{EN}_{SQ_{M-1,M_C}}})$ $\mathrm{EN}_{S_{M-1,M_C}}$
Photosensing
PE
$(V_{SQ_{1,1}})$ $V_{S_{1,1}}$
$(V_{SQ_{2,1}})$ $V_{S_{2,1}}$
$(V_{SQ_{3,1}})$ $V_{S_{3,1}}$
$(V_{SQ_{M,1}})$ $V_{S_{M,1}}$
$(\overline{\mathrm{EN}_{SQ_{1,2_R}}})$ $\mathrm{EN}_{S_{1,2_R}}$
$(V_{SQ_{1,2}})$ $V_{S_{1,2}}$
$(V_{SQ_{2,2}})$ $V_{S_{2,2}}$
$(V_{SQ_{3,2}})$ $V_{S_{3,2}}$
$(V_{SQ_{M,2}})$ $V_{S_{M,2}}$
$(\overline{\mathrm{EN}_{SQ_{2,3_R}}})$ $\mathrm{EN}_{S_{2,3_R}}$
$(V_{SQ_{1,3}})$ $V_{S_{1,3}}$
$(V_{SQ_{2,3}})$ $V_{S_{2,3}}$
$(V_{SQ_{3,3}})$ $V_{S_{3,3}}$
$(V_{SQ_{M,3}})$ $V_{S_{M,3}}$
$(\overline{\mathrm{EN}_{SQ_{3,4_R}}})$ $\mathrm{EN}_{S_{3,4_R}}$
$(\overline{\mathrm{EN}_{SQ_{N-1,N_R}}})$ $\mathrm{EN}_{S_{N-1,N_R}}$
$(V_{SQ_{1,N}})$ $V_{S_{1,N}}$
$(V_{SQ_{2,N}})$ $V_{S_{2,N}}$
$(V_{SQ_{3,N}})$ $V_{S_{3,N}}$
$(V_{SQ_{M,N}})$ $V_{S_{M,N}}$

Figure 5. Simplified scheme of how the charge redistribution can be reconfigured in the proposed focal-plane array.

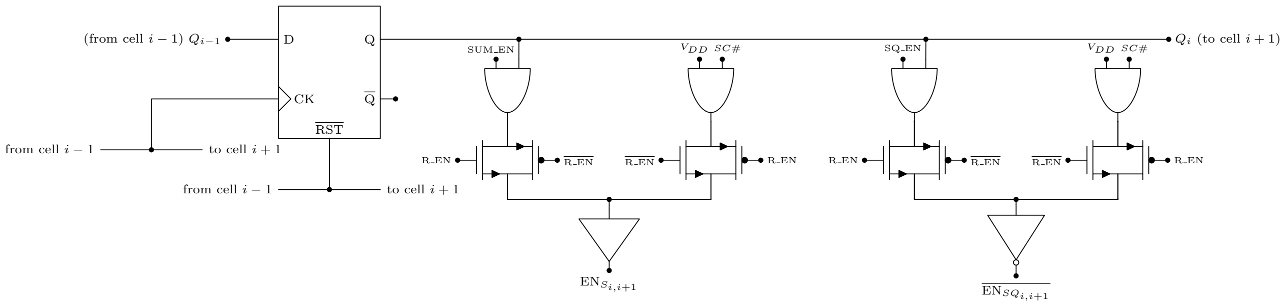

(from cell $i-1$) Q_{i-1}
D
Q
CK
$\overline{\mathrm{Q}}$
$\overline{\mathrm{RST}}$
from cell $i-1$
to cell $i+1$
from cell $i-1$
to cell $i+1$
SUM_EN
V_{DD} $SC\#$
SQ_EN
V_{DD} $SC\#$
Q_i (to cell $i+1$)
R_EN
$\overline{\mathrm{R\_EN}}$
$\overline{\mathrm{R\_EN}}$
R_EN
R_EN
$\overline{\mathrm{R\_EN}}$
$\overline{\mathrm{R\_EN}}$
R_EN
$\mathrm{EN}_{S_{i,i+1}}$
$\overline{\mathrm{EN}_{SQ_{i,i+1}}}$

Figure 6. Peripheral cell per column connection. The same cell is used per row connection in order to provide $\mathrm{EN}_{S_{j,j+1}}$ and $\overline{\mathrm{EN}_{SQ_{j,j+1}}}$.

4. POWER CONSUMPTION

The array just described features one of the main assets of the focal-plane sensing-processing approach: power efficiency. Bear in mind that we are targeting typical video frame rates, that is, around 30fps. At this rate, the switching power associated to the peripheral digital circuitry, measured by a magnitude of μW/MHz, will hardly impact energy consumption. Regarding the mixed-signal core, squaring and charge redistribution do not require extra energy once the initial voltages at the corresponding capacitors have been set. Thus, three major sources of power consumption are left: reset of the photodiode and sensing capacitance, precharge of the capacitance holding $V_{SQ_{ij}}$ and pixel copy operation. We will be considering a fixed frame rate of 30fps and a resolution of 320×240-px for the figures provided next. The reset of the photodiode is needed only once per frame. According to simulations, this operation demands 16.4pW per cell, 1.26μW for the whole array. Regarding precharge for subsequent pixel squaring, a single operation requires only 0.18pJ. However, it is performed at each cell of the array for each pixel of the square integral image at each scale. Adding up all these operations for five scales, the resulting power consumption is 42.4mW. Likewise, a single copy of pixel demands only 0.47pJ whereas all

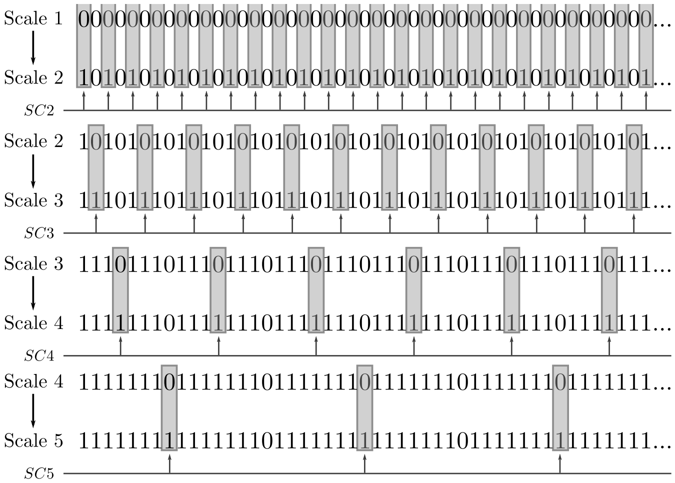


Figure 7. Distribution of the signals $SC\#$ in order to set five scales.

the operations required across five scales amount to 111.36mW. A total power consumption always less than 200mW is therefore expected for the mixed-signal processing array. As a reference, we can scale this figure in order to compare it to the power consumption reported for a smart camera handling 30×30-px images at 80fps:[15] 240mW . Under these conditions of image size and frame rate, the power consumption of our array boils down to less than 100μW. Of course this comparison is not fair enough since the camera described in[15] constitutes a general-purpose digital system carrying out the complete Viola-Jones processing flow. However, it still permits to give an idea of the energy efficiency reached by the approach presented. Starting at 100μW for imaging and low-level processing, it seems rather feasible to address the design of a vision system featuring a power consumption significantly less than 240mW.

5. CONCLUSIONS

We have demonstrated in this paper that the local descriptors required by the Viola-Jones framework are suitable for mixed-signal focal-plane implementation. The methodology and circuitry proposed to address such implementation has been described. We have also demonstrated the architectural advantages of our approach: exploitation of focal-plane distributed memory and ultra-low-power operation. The reconfigurability of the array at the core of our proposal endows the resulting processing lattice with additional low-level functionalities useful for different vision algorithms.

ACKNOWLEDGMENTS

This work is funded by MEyC (Spain) through projects TEC2012-38921-C02-01, co-funded by the European Regional Development Fund, and IPT-2011-1625-430000, by the Office of Naval Research (USA) through grant N000141110312, and by CDTI (Spain), co-funded by the European Regional Development Fund, through Project IPC-20111009.

REFERENCES

[1] Tuytelaars, T. and Mikolajczyk, K., "Local invariant feature detectors: A survey," *Foundat. Trends Comput. Graph. Vis.* **3**(3), 177–280 (2008).

[2] Zarándy, A., ed., [*Focal-plane Sensor-Processor Chips*], Springer (2011).

[3] Tomasi, C., [*Early Vision*], John Wiley & Sons, Ltd (2006).

[4] Unger, S., "A computer oriented toward spatial problems," *Proceedings of the IRE* **46**(10), 1744–1750 (1958).

[5] Wyatt, J., Keast, C., Seidel, M., Standley, D., Horn, B., Knight, T., Sodini, C., Lee, H.-S., and Poggio, T., "Analog VLSI systems for image acquisition and fast early vision," *Int. J. of Computer Vision* **8**(3), 217–230 (1992).

[6] Gottardi, M., Massari, N., and Jawed, S., "A 100μW 128×64 pixels contrast-based asynchronous binary vision sensor for sensor networks applications," *IEEE J. Solid-State Circuits* **44**(5), 1582–1592 (2009).

[7] Leñero Bardallo, J., Serrano-Gotarredona, T., and Linares-Barranco, B., "A 3.6μs latency asynchronous frame-free event-driven dynamic-vision-sensor," *IEEE J. Solid-State Circuits* **46**(6), 1443–1455 (2011).

[8] Fernández-Berni, J. and Carmona-Galán, R., "All-MOS implementation of RC networks for time-controlled Gaussian spatial filtering," *Int. J. of Circuit Theory and Applications* **40**(8), 859–876 (2012).

[9] Cottini, N., Gottardi, M., Massari, N., Passerone, R., and Smilansky, Z., "A 33μW 64×64 pixel vision sensor embedding robust dynamic background subtraction for event detection and scene interpretation," *IEEE J. Solid-State Circuits* **48**(3), 850–863 (2013).

[10] Oliveira, F., Haas, H., Gomes, J., and Petraglia, A., "CMOS imager with focal-plane analog image compression combining DPCM and VQ," *IEEE Trans. Circuits Syst. I* **60**(5), 1331–1344 (2013).

[11] Rodríguez-Vázquez, A., Domínguez-Castro, R., Jímenez-Garrido, F., Morillas, S., Listán, J., Alba, L., Utrera, C., Espejo, S., and Romay, R., "The Eye-RIS CMOS vision system," in [*Sensors, Actuators and Power Drivers; Integrated Power Amplifiers from Wireline to RF; Very High Frequency Front Ends*], Casier, H., Steyaert, M., and Roermund, A., eds., *Analog Circuit Design*, ch. 2, Springer (2008).

[12] Viola, P. and Jones, M., "Robust real-time face detection," *Int. J. of Computer Vision* **57**(2), 137–154 (2004).

[13] Acasandrei, L. and Barriga, A., "FPGA implementation of an embedded face detection system based on LEON3," in [*World Congress in Computer Science, Computer Engineering and Applied Computing*], (2012).

[14] Jia, H., Zhang, Y., Wang, W., and Xu, J., "Accelerating Viola-Jones face detection algorithm on GPUs," in [*IEEE Int. Conf. on Embedded Software and Systems*], 396–403 (2012).

[15] Camilli, M. and Kleihorst, R., "Demo: Mouse sensor networks, the smart camera," in [*5th ACM/IEEE Int. Conf. on Distributed Smart Cameras*], (2011).

[16] Acasandrei, L. and Barriga-Barros, A., "Accelerating Viola-Jones face detection for embedded and SoC environments," in [*5th ACM/IEEE Int. Conf. on Distributed Smart Cameras*], (2011).

[17] Mallat, S., "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(7), 674–693 (1989).

[18] Bradski, G., "The OpenCV Library," *Dr. Dobb's Journal of Software Tools* (2000). http://opencv.org/.

[19] Fernández-Berni, J., Carmona-Galán, R., and Carranza-González, L., "FLIP-Q: A QCIF resolution focal-plane array for low-power image processing," *IEEE J. Solid-State Circuits* **46**(3), 669–680 (2011).

[20] Fernández-Berni, J., Carmona-Galán, R., and Rodríguez-Vázquez, A., "Image filtering by reduced kernels exploiting kernel structure and focal-plane averaging," in [*IEEE European Conf. on Circuit Theory and Design (ECCTD)*], 229–232 (2011).

[21] Fernández-Berni, J., Carmona-Galán, R., and Rodríguez-Vázquez, A., "Reconfigurable focal-plane hardware for block-wise intra-frame HDR imaging," in [*Int. Image Sensor Workshop*], 289–292 (2013).