Image Feature Extraction Acceleration

Jorge Fernández-Berni, Manuel Suárez, Ricardo Carmona-Galán, Víctor M. Brea, Rocío del Río, Diego Cabello and Ángel Rodríguez-Vázquez

Abstract Image feature extraction is instrumental for most of the best-performing algorithms in computer vision. However, it is also expensive in terms of computational and memory resources for embedded systems due to the need of dealing with individual pixels at the earliest processing levels. In this regard, conventional system architectures do not take advantage of potential exploitation of parallelism and distributed memory from the very beginning of the processing chain. Raw pixel values provided by the front-end image sensor are squeezed into a high-speed interface with the rest of system components. Only then, after deserializing this massive dataflow, parallelism, if any, is exploited. This chapter introduces a rather different approach from an architectural point of view. We present two Application-Specific Integrated Circuits (ASICs) where the 2-D array of photo-sensitive devices featured by regular imagers is combined with distributed memory supporting concurrent processing. Custom circuitry is added per pixel in order to accelerate image feature extraction right at the focal plane. Specifically, the proposed sensing-processing chips aim at the acceleration of two flagships algorithms within the computer vision community:

J. Fernández-Berni (✉) · R. Carmona-Galán · R. del Río · Á. Rodríguez-Vázquez
Institute of Microelectronics of Seville (CSIC - Universidad de Sevilla),
C/ Américo Vespucio s/n, 41092 Seville, Spain
e-mail: berni@imse-cnm.csic.es

R. Carmona-Galán
e-mail: rcarmona@imse-cnm.csic.es

R. del Río
e-mail: rocio@imse-cnm.csic.es

Á. Rodríguez-Vázquez
e-mail: angel@imse-cnm.csic.es

V.M. Brea · M. Suárez · D. Cabello
Centro de Investigación en Tecnoloxías da Información (CITIUS),
University of Santiago de Compostela, Santiago de Compostela, Spain
e-mail: victor.brea@usc.es

D. Cabello
e-mail: diego.cabello@usc.es

A.I. Awad and M. Hassaballah (eds.), *Image Feature Detectors and Descriptors*,
Studies in Computational Intelligence 630, DOI 10.1007/978-3-319-28854-3_5

the Viola-Jones face detection algorithm and the Scale Invariant Feature Transform (SIFT). Experimental results prove the feasibility and benefits of this architectural solution.

Keywords Image feature extraction · Focal-plane acceleration · Distributed memory · Parallel processing · Viola-Jones · SIFT · Vision chip

1 Introduction

1.1 Embedded Vision

Embedded vision market is forecast to experience a notable and sustained growth during the next few years [1]. The integration of hardware and software technologies is reaching the required maturity to support this growth. At hardware level, the ever-increasing computational power of Digital Signal Processors (DSPs), Field Programmable Gate Arrays (FPGAs), General-Purpose Graphics Processing Units (GP-GPUs) and vision-specific co-processors permit to address the challenging processing requirements usually demanded by embedded vision applications [2]. At software level, the development of standards like OpenCL [3] or OpenVX [4] as well as tools like OpenCV [5] or CUDA [6] allow for rapid prototyping and shorter time to market.

A noticeable trend within this ecosystem of technologies is hardware parallelization, commonly in terms of processing operations [7, 8]. However, improving performance is not only a matter of parallelizing computational tasks. Memory management and dataflow organization are crucial aspects to take into account [9, 10]. In the case of memory management, the limitation arises from the so-called memory gap [11], leading to a substantial amount of idle time for processing resources due to slow memory access. The influence of a well-designed dataflow organization on the system performance is intimately related to this limitation. The overall objective must be to avoid moving large amounts of information pieces back and forth between system components via intermediate memory modules [2]. Optimization on this point must be planned after a comprehensive analysis of the processing flow featured by the targeted algorithm [9]. Particularly, early vision involving pixel-level operations must be carefully considered as it normally constitutes the most demanding stage in terms of processing and memory resources.

1.2 Focal-Plane Sensing-Processing Architecture

When all these key factors shaping performance are closely examined from an architectural point of view, a major disadvantage of conventional system architectures becomes evident. As can be observed in Fig. 1, vision systems typically consist of a

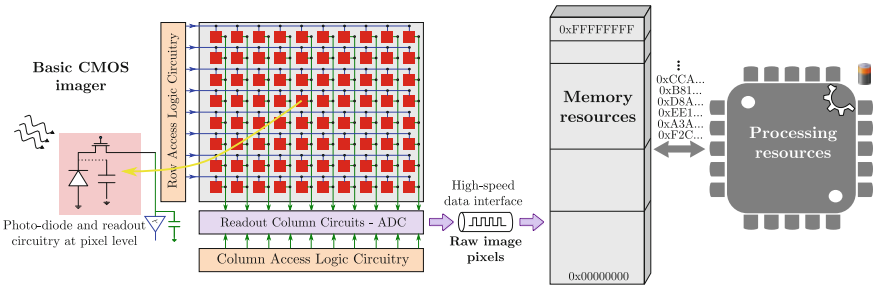

Fig. 1 Conventional architecture of embedded vision systems: image sensor, high-speed Analog-to-Digital Conversion (ADC), memory and processing resources (DSP, GPU etc.)

front-end imager delivering high-quality images at high speed to the rest of system components. This arrangement, by itself, generates a critical bottleneck associated with the huge amount of raw data rendered by the imager that must be subsequently stored and processed from scratch. But even more importantly, it precludes a first stage of processing acceleration from taking place just at the focal plane in a distributed and parallel way. Notice that the imager inevitably requires the physical realization of a 2-D array of photo-sensitive devices topographically assigned to their corresponding pixel values. This array can be exploited as distributed memory where the data are directly accessible for concurrent processing by including suitable circuitry at pixel level. As a result, the imager will be delivering pre-processed images, possibly in addition to the original raw information in case the algorithm needs it to superpose the processing outcome—e.g. highlighting the location of a tracked object. This architectural approach, referred in the literature as *focal-plane sensing-processing* [12] and represented in Fig. 2, presents two fundamental advantages when compared to that of Fig. 1. First of all, it enables a drastic reduction of memory accesses during low-level processing stages, where pixel-wise operations are common. Secondly, it permits to design ad-hoc circuitry to accelerate a vision

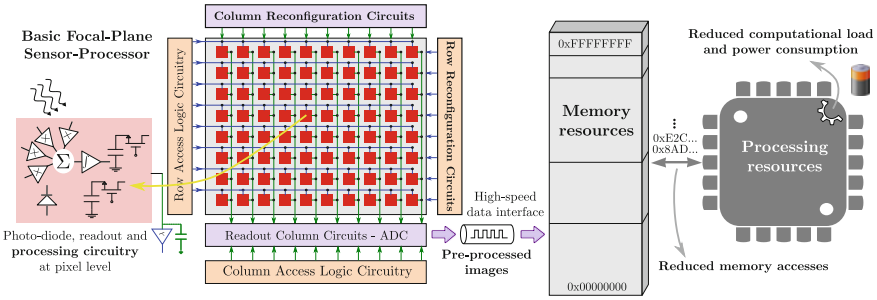

Fig. 2 Proposed architecture for focal-plane acceleration of image feature extraction. The pixel array is exploited as distributed memory including per-pixel circuitry for parallel processing

algorithm according to its specific characteristics. This circuitry can even be implemented in the analog domain for the sake of power and area efficiency since the pixel values at the focal plane have not been converted to digital yet. On the flip side, the incorporation of processing circuitry at pixel level reduces, for a prescribed pixel area, the sensitivity of the imager as less area is devoted to capture light. This drawback could be overcome by means of the so-called 3-D integration technologies [13, 14]. In this case, a sensor layer devoting most of its silicon area to capture light would be stacked and vertically interconnected onto one or more layers exclusively dedicated to processing. While not mature enough yet for reliable implementation of sensing-processing stacks, 3-D manufacturing processes will most surely boost the application frameworks of the research hereby presented.

All in all, this chapter introduces two full-custom focal-plane accelerator sensing-processing chips. They are our first prototypes aiming respectively at speeding up the image feature extraction of two flagships algorithms within the embedded vision field: the Viola-Jones face detection algorithm [15] and the Scale Invariant Feature Transform (SIFT) [16]. To the best of our knowledge, no prior attempts pointing to these algorithms have been reported for the proposed sensing-processing architectural solution. The chapter is organized as follows. After briefly describing both algorithms, we justify the operations targeted for implementation at the focal plane. We demonstrate that these operations feature a common underlying processing primitive, the Gaussian filtering, convenient for pixel-level circuitry. We then explain how this processing primitive has been implemented on both chips. Finally, we provide experimental results and discuss the guidelines of our future work on this subject matter.

2 Vision Algorithms

2.1 Viola-Jones Face Detection Algorithm

The Viola-Jones sliding window face detector [15] is considered a milestone in real-time generic object recognition. It requires a cumbersome previous training, demanding a large number of cropped frontal face samples. But once trained, the detection stage is fast thanks to the computation of the integral image, an intermediate image representation speeding up feature extraction, and to a cascade of classifiers of progressive complexity. A basic scheme of the Viola-Jones processing flow is depicted in Fig. 3. Despite its simplicity and detection effectiveness, the algorithm still requires a considerable amount of computational and memory resources in terms of embedded system affordability. Different approaches have been proposed in the literature in order to increase the implementation performance: by exploiting the highly parallel computation structure of GPUs [17, 18]; by making the most of the logic and memory capabilities of FPGAs [19, 20]; by custom design of specialized digital hardware [21] etc.

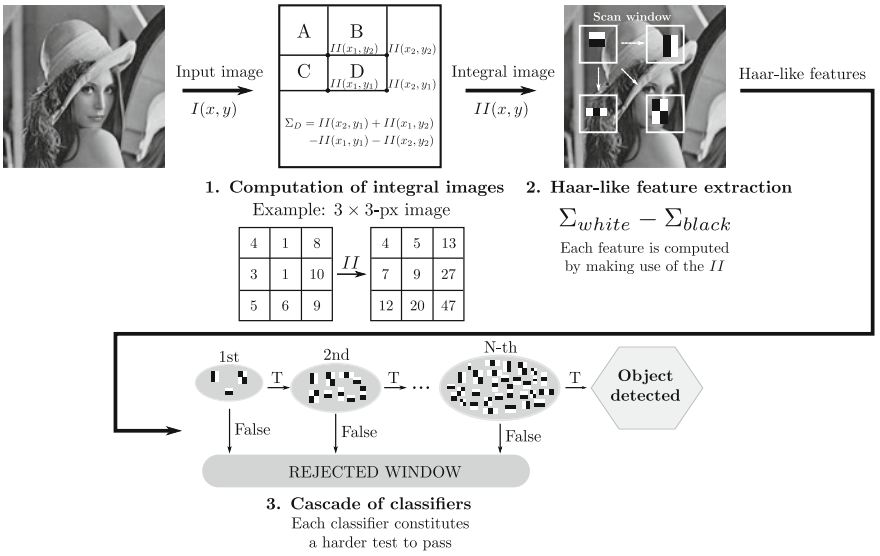

Fig. 3 Simplified scheme of the Viola-Jones processing flow

In order to evaluate the possibilities for focal-plane acceleration, our interest focuses on pixel-level operations. For the Viola-Jones algorithm, these operations take place during the computation of the integral image, defined as:

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

where $I(x, y)$ represents the input image. That is, each pixel composing $II(x, y)$ is equal to the sum of all the pixels above and to the left of the corresponding pixel at the input image. The first advantage of the integral image is that its calculation permits to compute the sum of any rectangular region of the input image by accessing only four pixels of the matrix $II(x, y)$. This is critical for real-time operation, given the potential large number of Haar-like features to be extracted—2135 in total for the OpenCV baseline implementation. The second advantage is that the computation of the integral image fits very well into a pipeline architecture—typically implemented in DSPs—by making use of the following pair of recurrences:

$$\begin{cases} r(x, y) = r(x, y - 1) + I(x, y) \\ II(x, y) = II(x - 1, y) + r(x, y) \end{cases} \tag{2}$$

with $r(x, 0) = 0$ and $II(0, y) = 0$. The matrix $II(x, y)$ can thus be obtained in one pass over the input image.

Despite these advantages, the purely sequential approach defined by Eq. (2) is still computationally expensive and memory access intensive [20, 22]. It usually

accounts for a large fraction of the total execution time due to its linear dependence on the number of pixels of the input image [23]. Thus, its parallelization would boost the performance of the whole algorithm. In the next sections, we will propose an acceleration scheme that can clearly benefit from the concurrent operation and distributed memory provided by focal-plane architectures.

2.2 *Scale Invariant Feature Transform (SIFT)*

The SIFT algorithm constitutes a combination of keypoint detector and corresponding feature descriptor encoding [16]. It can be broken up into four main steps:

1. Scale-space extrema detection: generation of the Gaussian and subsequent Difference-of-Gaussian (DoG) pyramids, searching for the extrema points in the DoG pyramid.
2. Accurate keypoint location in the scale space.
3. Orientation assignment to the corresponding keypoint, searching for the main orientation or main component from the gradient in its neighborhood.
4. Keypoint descriptor: construction of a vector representative of the local characteristics of the keypoint in a wider neighborhood with orientation correction.

Numerous examples of SIFT implementations on different platforms have been reported: general-purpose CPU [24], GPU [25, 26], FPGA [27, 28], FPGA + DSP [29], specific digital co-processors [30] etc. As for the Viola-Jones, the lowest-level operation of the SIFT, namely the generation of the Gaussian pyramid, dominates the workload of the algorithm, reaching up to 90 % of the whole process [31]. Figure 4

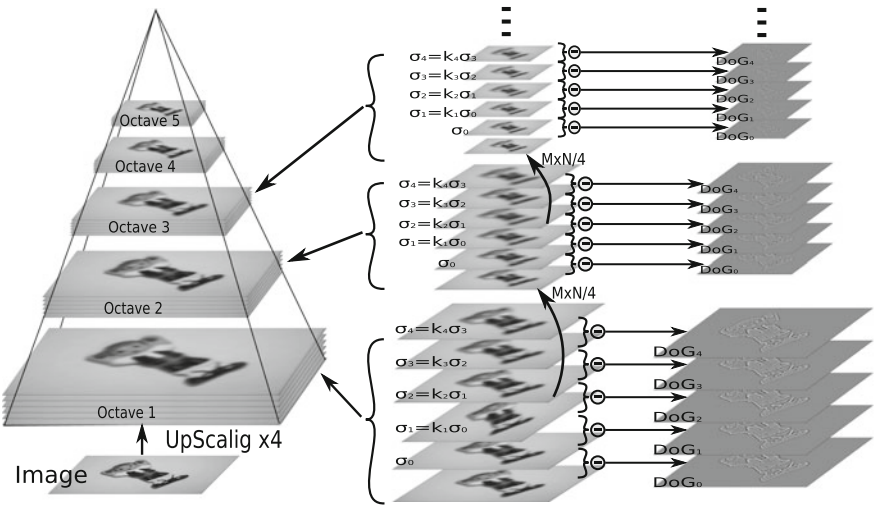

Fig. 4 Gaussian pyramid with its associated DoGs

shows an example of Gaussian pyramid with its associated DoGs. It is made up of sets of filtered images (scales). Every octave starts with a half-sized downscaling of the previous octave. The filter bandwidth, σ, applied for a new scale within each octave is the one applied in the previous scale multiplied by a constant factor k: $\sigma_n = k\sigma_{n-1}$. Each octave is originally divided into an integer number of scales, s, so $k = 2^{1/s}$. A total of $s + 3$ [16] images must be produced in the stack of blurred images for extrema detection to cover a complete octave. Once the Gaussian pyramid is built, the scales are subtracted from each other, obtaining the DoGs as an approximation to the Laplacian operator.

Our objective is therefore to accelerate the SIFT Gaussian pyramid generation by means of in-pixel circuitry performing concurrent processing. For the sake of relaxation on the hardware requirements, we carried out a preliminary study to determine the number of octaves and scales to be provided by our focal-plane sensor-processor. For this study, we used a publicly available version of SIFT in MATLAB [32]. Every octave is generated from a scale of the previous octave downsized by a 1/4 factor (1/2 × 1/2), decreasing the pixels per octave. Therefore, the maximum potential keypoints decrease rapidly with the octaves $o = 0, 1, 2 \ldots$ as $M \times N/2^{o \times 2}$, with $M \times N$ being the size of the input image. Assuming a resolution of 320 × 240 pixels (QVGA), we obtained the keypoints for two images under many scales and rotation transformations. The reason of this moderate resolution is that the area to be allocated for in-pixel processing circuitry makes it difficult to reach larger resolutions in standard CMOS technologies with a reasonable chip size. The results for two of the applied transformations together with the test images are represented in Fig. 5. Clearly, the 3 first octaves render almost all the keypoints. Concerning scales, we have two opposite contributions. On the one hand, less scales per octave means more distance between scales, causing more pixels to exceed the threshold to be sorted out as keypoints. On the other hand, reducing scales also means to diminish the total number of potential keypoints. Both combined effects make it difficult to choose a specific value for scales as in the case of the octaves. The result of the scale analysis for the same respective test images and transformations as in Fig. 5 is depicted

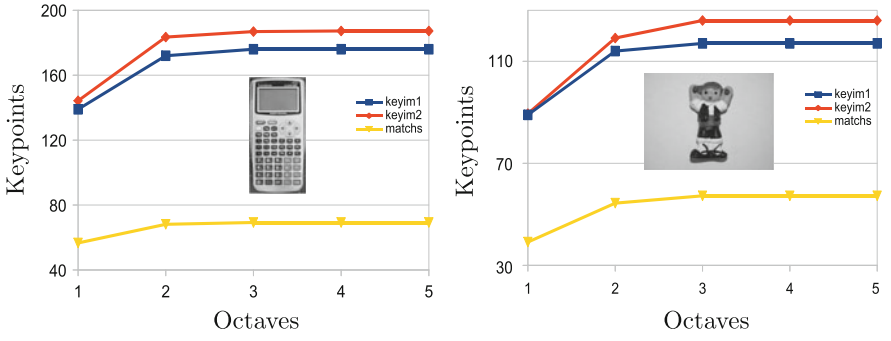

Fig. 5 The number of keypoints hardly increases from the 3 first octaves. This will be the reference value for our implementation

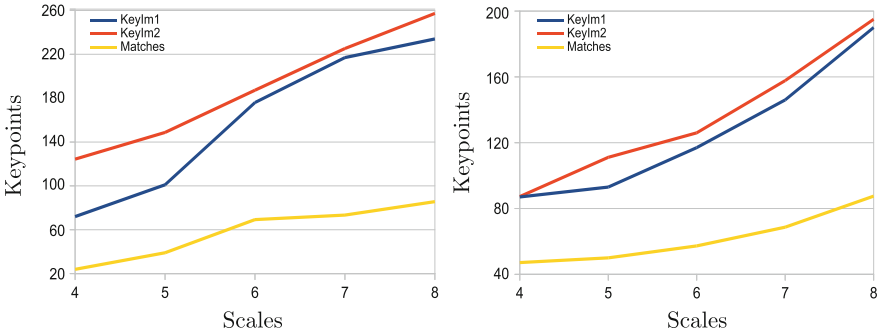

Fig. 6 The number of keypoints increases with the number of scales per octave. Trading this result for computational demand and hardware complexity leads to a targeted number of 6 scales. The same test images and transformations as in Fig. 5 are respectively used

in Fig. 6. It shows that the amount of keypoints increases monotonically with the scales. Nevertheless, increasing the scales per octave is not an option because of its corresponding computational demand and hardware complexity. Trading all these aspects, we conclude that 6 scales suffice for Gaussian pyramid generation at the focal plane. This figure coincides with the number of scales proposed in [16].

3 Gaussian Filtering

We demonstrate in this section that Gaussian filtering is the common underlying processing primitive for both the Viola-Jones and SIFT algorithms. While the role of Gaussian filtering is well defined for the latter, it is not obvious at all for the former. In order to understand the relation, we first need to establish a formal mathematical framework. Gaussian filtering is best illustrated in terms of a diffusion process. The concept of diffusion is widely applied in physics. It explains the equalization process undergone by an initially uneven concentration of a certain magnitude. A typical example is heat diffusion. Mathematically, a diffusion process can be defined by considering a function $V(\mathbf{x}, t)$ defined over a continuous space, in this case a plane, for every time instant. At each point $\mathbf{x} = (x_1, x_2)$, the linear diffusion of the function $V(.)$ is described by the following well-known partial differential equation [33]:

$$\frac{\partial V}{\partial t} = \nabla \cdot (D \nabla V) \tag{3}$$

where D is referred to as the diffusion coefficient. If D does not depend on the position:

$$\frac{\partial V}{\partial t} = D \nabla^2 V \tag{4}$$

and realizing the spatial Fourier transform of this equation, we obtain:

$$\frac{\partial \hat{V}(\mathbf{k})}{\partial t} = -4\pi^2 D|\mathbf{k}|^2 \hat{V}(\mathbf{k}) \tag{5}$$

where $\mathbf{k}$ represents the wave number vector in the continuous Fourier domain. Finally, by solving this equation we have:

$$\hat{V}(\mathbf{k}, t) = \hat{V}(\mathbf{k}, 0) e^{-4\pi^2 Dt|\mathbf{k}|^2} \tag{6}$$

where $\hat{V}(\mathbf{k}, t)$ is the spatial Fourier transform of the function $V(.)$ at time instant t and $\hat{V}(\mathbf{k}, 0)$ is the spatial Fourier transform of the function $V(.)$ at time $t = 0$, that is, just before starting the diffusion. Equation (6) can be written as a transfer function:

$$\hat{G}(\mathbf{k}, t) = \frac{\hat{V}(\mathbf{k}, t)}{\hat{V}(\mathbf{k}, 0)} = e^{-4\pi^2 Dt|\mathbf{k}|^2} \tag{7}$$

which, by defining $\sigma = \sqrt{2Dt}$, is transformed into:

$$\hat{G}(\mathbf{k}, \sigma) = e^{-2\pi^2 \sigma^2 |\mathbf{k}|^2} \tag{8}$$

This transfer function corresponds to the Fourier transform of a spatial Gaussian filter of the form:

$$G(\mathbf{x}, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{|\mathbf{x}|^2}{2\sigma^2}} \tag{9}$$

and therefore the diffusion process is equivalent to the convolution expressed by the following equation:

$$V(\mathbf{x}, t) = \frac{1}{2\pi\sigma^2} e^{-\frac{|\mathbf{x}|^2}{2\sigma^2}} * V(\mathbf{x}, 0) \tag{10}$$

We can see that a diffusion process intrinsically entails a spatial Gaussian filtering which takes place along time. The width of the filter is determined by the time the diffusion is permitted to evolve: the longer the diffusion time, t, the larger the width of the corresponding filter, σ. This means that, ideally, any width is possible provided that a sufficiently fine temporal control is available. From the point of view of the Fourier domain, we can define the diffusion as an isotropic lowpass filter whose bandwidth is controlled by t. The longer t, the narrower the bandwidth of the filter around the dc component (Fig. 7). Eventually, for $t \to \infty$, all the spatial frequencies but the dc component are removed. Furthermore, this dc component is completely unaffected by the diffusion, that is, $\hat{G}(\mathbf{0}, t) = 1 \ \forall t$. It is just this characteristic of the Gaussian filtering what constitutes the missing link with the computation of the integral image. When discretized and applied to a set of pixels, this property says that a progressive Gaussian filtering eventually leads to the average of the values the

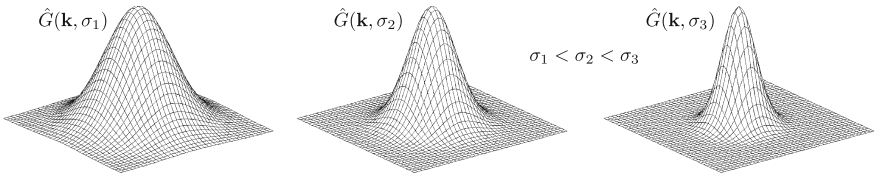

Fig. 7 Spatial Gaussian filters with increasing σ represented in the Fourier domain

pixels had before starting the filtering process. This average is a scaled version of the sum of the original pixels, precisely the calculation required for each pixel of the integral image. Furthermore, as we will see shortly, the averaging process inherent to the Gaussian filtering is extremely helpful to cope at hardware level with the large signal range demanded by the computation of the integral image.

4 Focal-Plane Implementation of Gaussian Filtering

The simple circuit depicted in Fig. 8a is our starting point to explain how we have addressed the design of in-pixel circuitry capable of implementing Gaussian filtering. Assuming that the initial conditions of the capacitors are V_{10} and V_{20}, the evolution of the circuit dynamics is described by:

$$\begin{cases} C\frac{dV_1}{dt} = -\frac{V_1(t)-V_2(t)}{R} \\ C\frac{dV_2}{dt} = \frac{V_1(t)-V_2(t)}{R} \end{cases} \tag{11}$$

whose solution is:

$$\begin{bmatrix} V_1(t) \\ V_2(t) \end{bmatrix} = \frac{1}{2}(V_{10}+V_{20})\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \frac{1}{2}(V_{10}-V_{20})\begin{bmatrix} 1 \\ -1 \end{bmatrix} e^{-2t/\tau} \tag{12}$$

where $\tau = RC$. Equation (12) physically represents a charge diffusion process—i.e. Gaussian filtering—taking place along time between both capacitors at a pace deter-

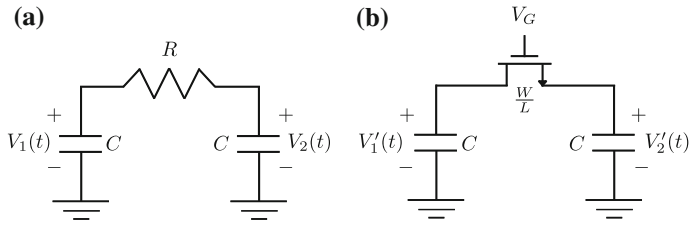

Fig. 8 2-node ideal diffusion circuit (**a**) and its transistor-based implementation (**b**)

mined by the time constant τ. For $t \to \infty$, both capacitors hold the same voltage, $(V_{10} + V_{20})/2$, that is, the average of their initial conditions. In order to achieve an area-efficient physical realization of this circuit, we can substitute the resistor by an MOS transistor (Fig. 8b) whose gate terminal additionally permits to control the activation-deactivation of the dynamics described by Eq. (12). Now suppose that V_{10} and V_{20} correspond respectively to two neighboring pixel values resulting from a photo-integration period previously set to capture a new image. If you are meant to compute the integral image from this new image, you will eventually want to add up both pixels as fast as possible. This can be accomplished by designing the proposed circuit with the minimum possible time constant τ in order to rapidly reach the steady state. Conversely, if you are meant to obtain the Gaussian pyramid, you will need fine control of the filtering process in order to increasingly blur the just captured image. In this case, the time constant τ cannot be arbitrarily small for the sake of making that fine control feasible. There are therefore conflicting design requirements depending on the specific task to be implemented by our basic circuit. In this scenario, we next present the particular realization of the diffusion process satisfying such requirements for both, the integral image computation and the Gaussian pyramid generation.

4.1 Focal-Plane Circuitry for Integral Image

A simplified scheme of how the charge diffusion process just described can be generalized for a complete image is depicted in Fig. 9. The MOS transistor in Fig. 8b has been substituted, to avoid clutter, for a simple switch for each connection between neighboring pixels in horizontal and vertical directions. This also highlights the fact that the MOS transistors are designed to have the minimum possible resistance when they are set ON, thus contributing to reduce the time constant τ. The state of these switches—ON or OFF—is controlled by the reconfiguration signals $\mathrm{EN}_{S_{i,i+1_C}}$ and $\mathrm{EN}_{S_{j,j+1_R}}$ for columns and rows respectively. The voltages $V_{px_{i,j}}$ represent the analog pixel values just after the photo-diode array has captured a new image. The integral image—really the averaged version provided by the diffusion process—is obtained by progressively establishing the adequate interconnection patterns in $\mathrm{EN}_{S_{i,i+1_C}}$ and $\mathrm{EN}_{S_{j,j+1_R}}$ according to the location of the pixel $II(x, y)$ being calculated at the moment. For example, we would need to activate $\mathrm{EN}_{S_{1,2_C}}$ and $\mathrm{EN}_{S_{1,2_R}}$, letting the remaining signals deactivated, in order to compute $II(2, 2)$. Each diffusion process producing an integral image pixel is followed by a stage of analog-to-digital conversion that takes place concurrently with the readjustment of the interconnection patterns for the next pixel to be computed. More details about the whole process and the additional circuitry required per pixel can be found in [34].

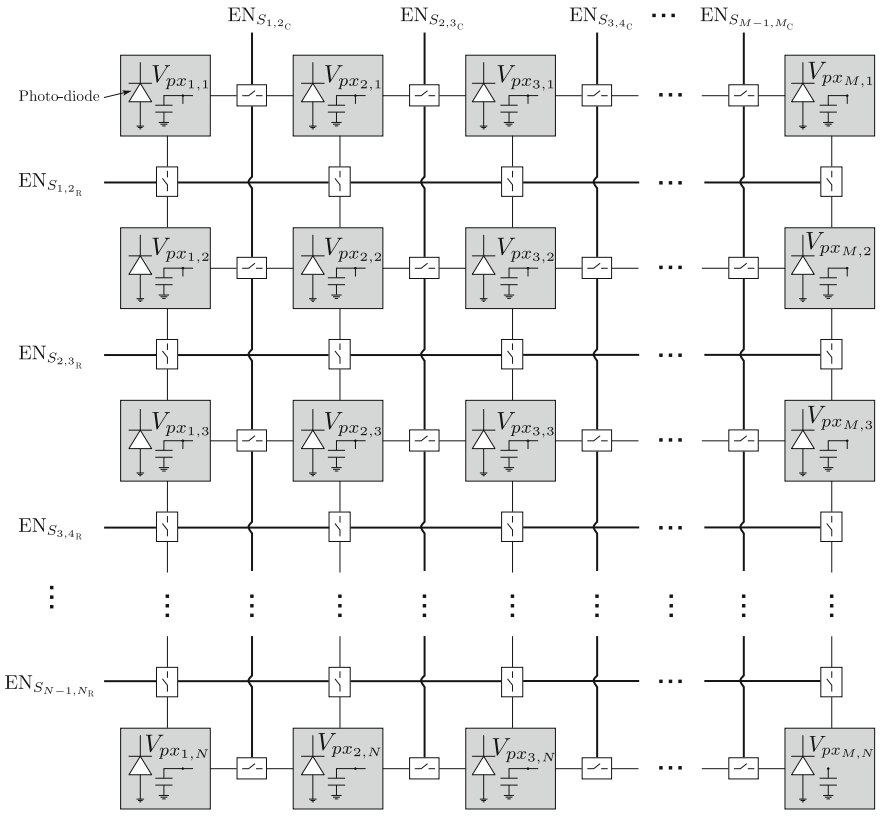

Fig. 9 Simplified scheme of how the diffusion process can be reconfigured to compute the integral image at the focal plane

4.2 Focal-Plane Circuitry for Gaussian Pyramid

As previously mentioned, the generation of the Gaussian pyramid requires an accurate control of the diffusion process in the circuit of Fig. 8b. A possible approach to achieve such control is to design specific on-chip circuitry providing precise timing over the gate signal of the MOS transistor [35]. Another possibility, featuring more linearity and even further diffusion control, is considered here. It is based on so-called Switched Capacitor (SC) circuits [36]. In this case, our reference circuit of Fig. 8a is transformed into that of Fig. 10. Two intermediate capacitors are introduced along with four switches enabling a gradual charge diffusion between the capacitors holding neighboring pixel values. Two non-overlapping clock phases driving the switches are used to carry out this progressive transfer of charge. It can be mathematically demonstrated [37] that this circuit configuration, called 'double Euler', is equivalent to apply a Gaussian filter with a width σ (see Eqs. (7) and (8)) given by:

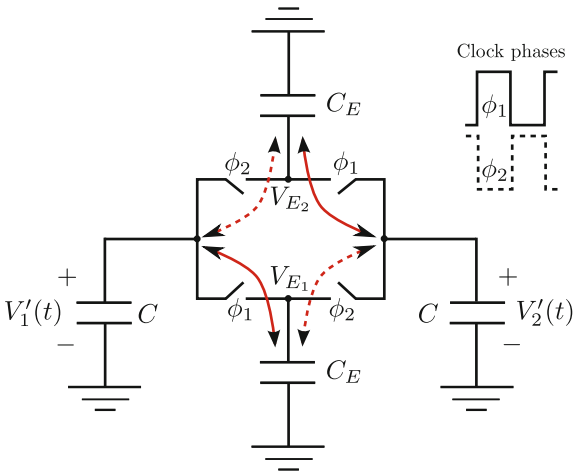

Fig. 10 In order to compute the Gaussian pyramid, two intermediate capacitors and four switches permit to gradually perform the charge diffusion between the capacitors holding neighboring pixel values

$$\sigma = \sqrt{\frac{2nC_E}{C}} \tag{13}$$

where n is the number of cycles completed by the clock phases. We are assuming that $C_E \ll C$. A simulation example of a four-pixel diffusion featuring a diffusion cycle as short as 90 ns is shown in Fig. 11. In the final physical realization, this diffusion

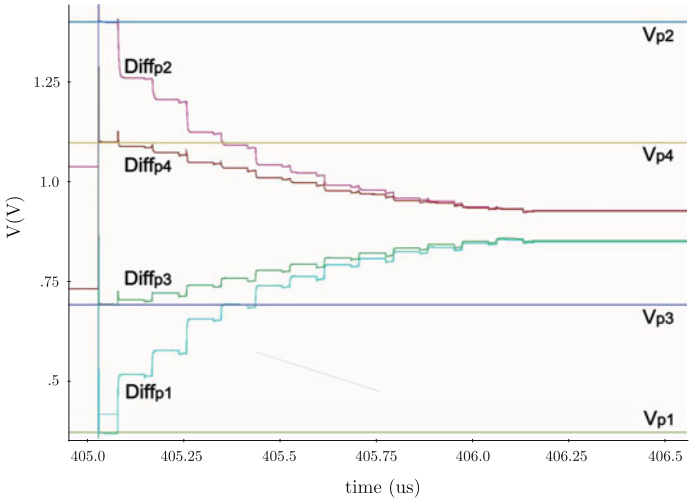

Fig. 11 Simulated temporal evolution of a four-pixel diffusion based on a double Euler SC circuit

cycle is adjusted in such a way that the targeted 3 octaves and 6 scales per octave can be attained. We therefore conclude that the discrete-time SC-based implementation of charge diffusion between capacitors provides the requested fine control of the underlying Gaussian filtering empowering the generation of the Gaussian pyramid at the focal plane.

5 Experimental Results

5.1 Viola-Jones Focal-Plane Accelerator Chip

The proposed prototype vision sensor presents the floorplan depicted in Fig. 12, featuring the elementary sensing-processing pixel shown in Fig. 13. The pixel array can be reconfigured block-wise by peripheral circuitry. The reconfiguration patterns are loaded serially into two shift registers that determine respectively which neighbor columns and rows can interact and which ones stay disconnected. There is also the possibility of loading in parallel up to six different patterns representing six successive image pixelation scales. This is achieved by means of control signals distributed regularly along the horizontal and vertical dimensions of the array [34]. The reconfiguration signals coming from the periphery map into the signals $\mathrm{EN}_{S_{i,i+1}}$, $\mathrm{EN}_{S_{j,j+1}}$, $\overline{\mathrm{EN}_{SQ_{i,i+1}}}$ and $\overline{\mathrm{EN}_{SQ_{j,j+1}}}$ at pixel level, where the coordinates (i, j) denote the location of the array cell considered. These signals control the activation of MOS switches for charge redistribution between the nMOS capacitors holding the voltages $V_{S_{ij}}$ and $V_{SQ_{ij}}$, respectively. Charge redistribution is the primary processing task that supports all the functionalities of the array, enabling low-power operation. Concerning A-to-D conversion, there are four 8-bit ADCs. These converters feature a tunable conversion range, including rail-to-rail, and a conversion time of 200 ns when clocked at 50 MHz. The column and row selection circuitry is also implemented by peripheral shift registers where a single logic '1' is shifted according to the location of the pixel to be converted.

The prototype chip together with the FPGA-based test system where it has been integrated can be seen in Fig. 14. An example of on-chip integral image computation is depicted in Fig. 15. As just explained, the sensing-processing array is capable of computing an averaged version of the actual integral image defined by Eq. (1), mathematically described as:

$$II_{av}(x, y) = \frac{1}{x \cdot y} \sum_{x'=1}^{x} \sum_{y'=1}^{y} I(x', y') \tag{14}$$

In Fig. 15, we can visualize the averaged integral image delivered by the chip and the integral image that can be directly derived from it. This integral image is compared with the ideal case obtained off-chip with MATLAB from the original image captured by the sensor, attaining an RMSE of 1.62 %. Notice that, in order

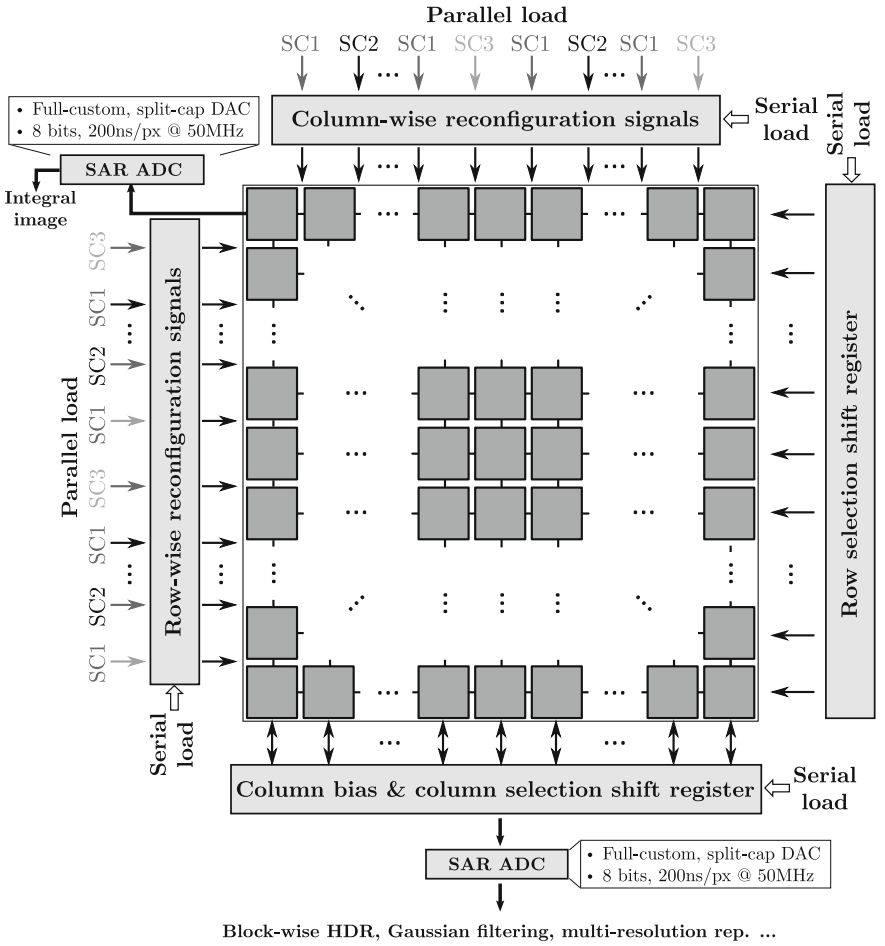

Fig. 12 Floorplan of the Viola-Jones focal-plane accelerator chip

to obtain $II(x, y)$, the only operation to be performed off-chip over $II_{av}(x, y)$ is to multiply each averaged pixel by its row and column number. No extra memory accesses are required for this task.

The chip has been manufactured in a standard 0.18 μm CMOS process. It features a resolution of 320×240 pixels and a power consumption of 55.2 mW when operating at 30 fps. This power consumption includes the image capture at that frame rate, the computation of the integral image for each captured image and the analog-to-digital conversion of the outcome for off-chip delivery. This figure is similar to that of state-of-the-art commercial image sensors, in this case with the add-on of focal-plane pre-processing alleviating the computational load of subsequent stages. The undesired effects of this add-on are reduced resolution and lower sensitivity. As mentioned in the introduction, these handicaps could be surmounted by 3-D integration.

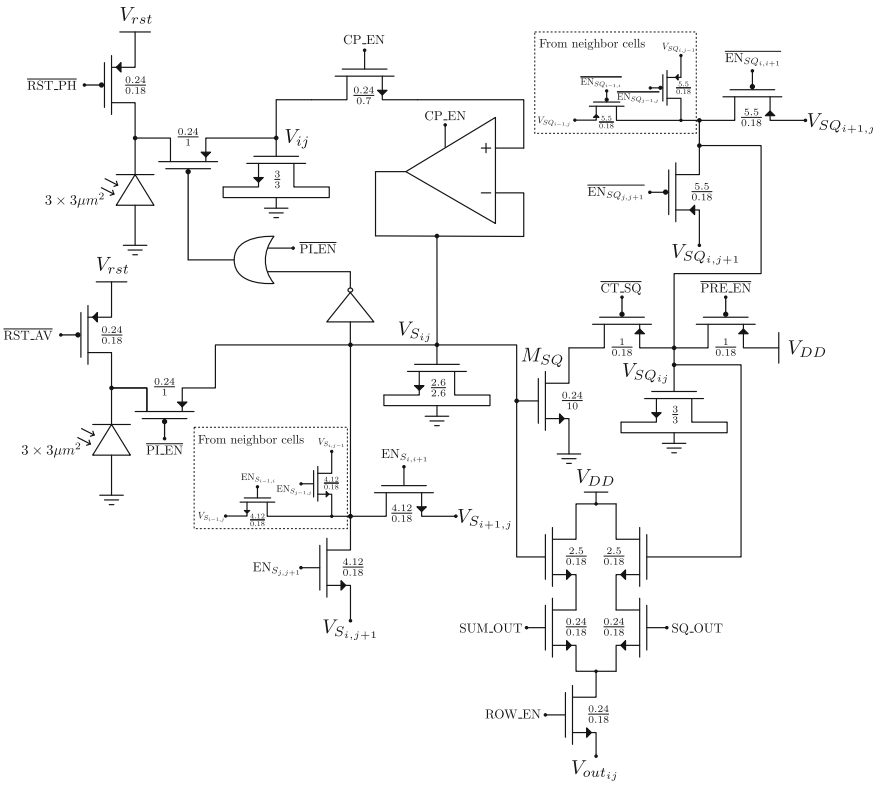

Fig. 13 Elementary sensing-processing pixel of the Viola-Jones focal-plane accelerator chip

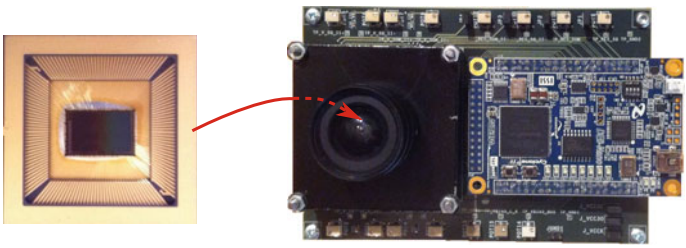

Fig. 14 Photograph of the Viola-Jones focal-plane accelerator chip and the FPGA-based system where it has been integrated

A direct transformation of the simplified scheme of Fig. 9 into a stacked structure is possible, as shown in Fig. 16. The top tier would exclusively include photo-diodes and some readout circuitry whereas the bottom tier would implement the reconfigurable diffusion network. The interconnection between both tiers would be carried out by the so-called Through-Silicon Vias (TSVs). This structure keeps maximum parallelism at processing while drastically increasing resolution and sensitivity.

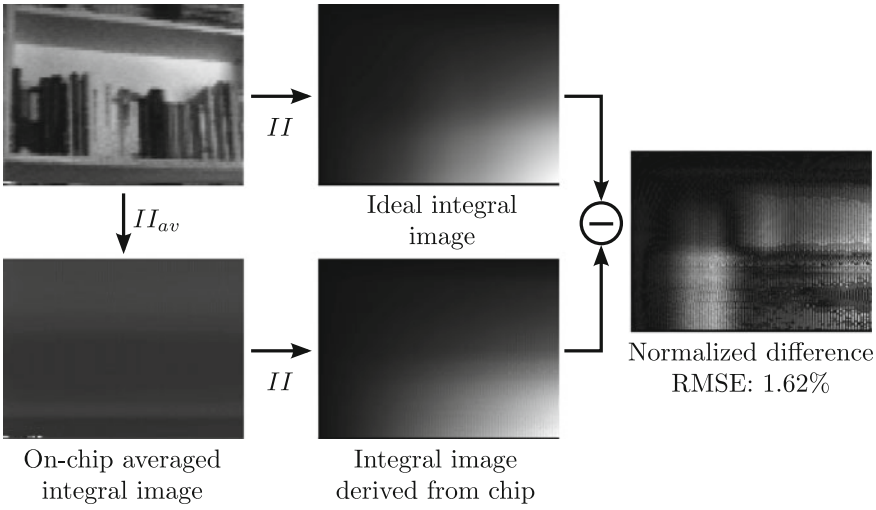

Fig. 15 Example of on-chip integral image computation and comparison with the ideal case

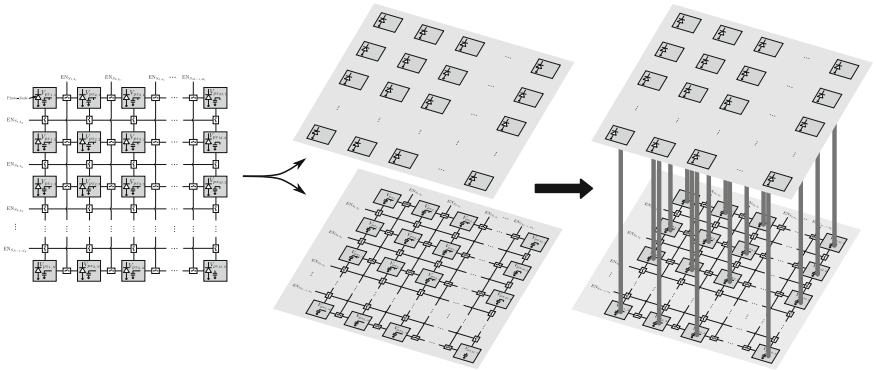

Fig. 16 Transformation of the simplified scheme of Fig. 9 into a stacked structure

5.2 SIFT Focal-Plane Accelerator Chip

The SIFT accelerator chip presents a similar floorplan to that of the Viola-Jones prototype. However, its elementary sensing-processing cell significantly differs. A simplified scheme is depicted in Fig. 17. The constituent blocks are mainly four photo-diodes, the local analog memories (LAMs), the comparator for A/D conversion and the switched capacitor network. During the acquisition stage, the photo-diodes, the capacitor C and the LAMs work together to implement a technique known as correlated double sampling [38] that improves the image quality. The LAMs jointly with the diffusion network carry out the Gaussian filtering. The capacitor C and the inverter make up the A/D comparator that would drive a register in the bottom

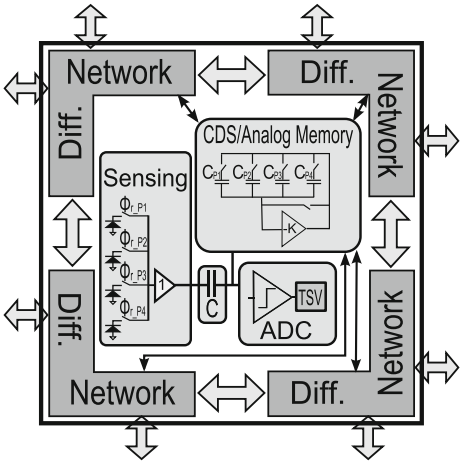

Fig. 17 Simplified scheme of the elementary sensing-processing cell designed for the SIFT focal-plane accelerator chip

tier by a TSV on CMOS-3D technologies, or peripheral circuits on conventional CMOS. Every cell is 4-connected to its closest neighbors in the North, South, East and West directions. Given that every cell includes four photo-diodes, 4 internal and 8 peripheral interconnections are required.

Two microphotographs of the chip together with the different components of the camera module built for test purposes are reproduced in Fig. 18. This prototype, also manufactured in a standard 0.18μm CMOS process, features a resolution of 176×120 pixels and can generate 120 Gaussian pyramids per second with a power consumption of 70 mW. One of the operations required for Gaussian pyramid generation is downscaling. As previously commented, the 3 first octaves are the most important ones in the performance of SIFT. This corresponds with downscaling at ratios 4:1 and 16:1 for octaves 2 and 3, respectively. The chip includes the hardware required to implement this spatial resolution reduction. An example is shown in Fig. 19. The images to the left are represented with the same sizes in order to visually highlight the effects of downscaling. Another example, in this case of on-chip Gaussian filtering, is shown in Fig. 20. The upper left image constitutes the input whereas the three remaining images, from left to right and top to down, correspond to $\sigma = 1.77$, (clock cyles $n = 19$), $\sigma = 2.17$ ($n = 29$), and $\sigma = 2.51$ ($n = 39$). More details about the performance of this chip can be found in [39].

This chip was conceived, from the very beginning, for implementation in 3-D integration technologies [40]. Unfortunately, these technologies are not mature enough yet for reliable fabrication. Manufacturing costs of prototypes are also extremely high for the time being, with long turnarounds, exceeding 1 year. In these circumstances, we were forced to redistribute the original two-tier circuit layout devised for a CMOS 3-D stack in order to fit it into a conventional planar CMOS technology. The result is depicted in Fig. 21.

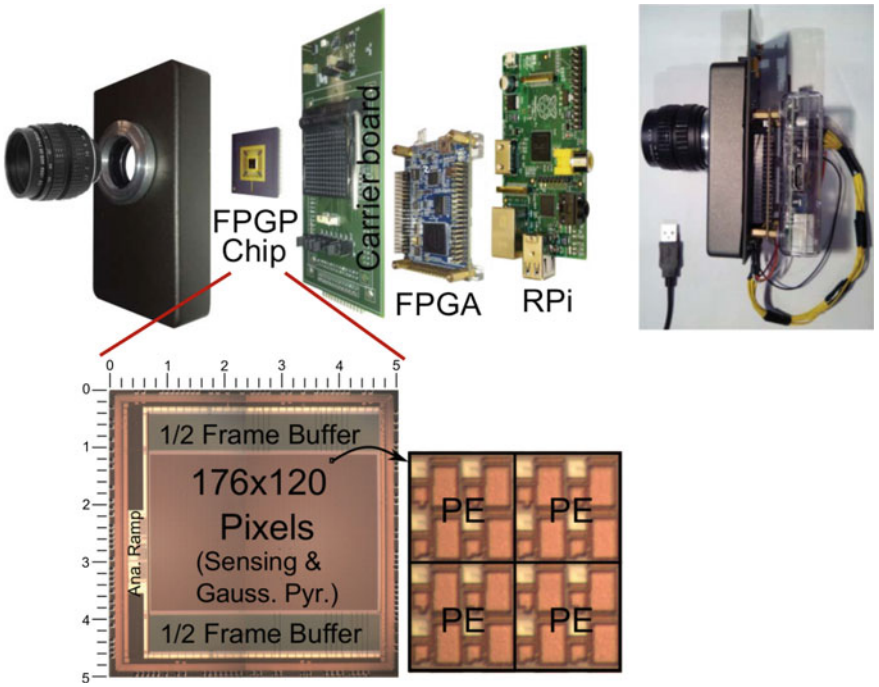

Fig. 18 Photograph of the SIFT focal-plane accelerator chip together with the camera module where it has been integrated

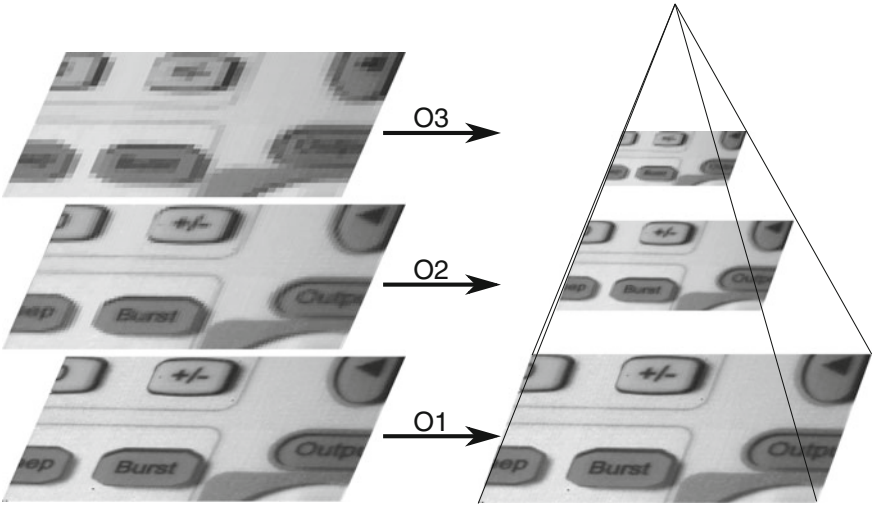

Fig. 19 On-chip image resolution reduction by 4:1 and 16:1 as part of the calculation of the pyramid octaves

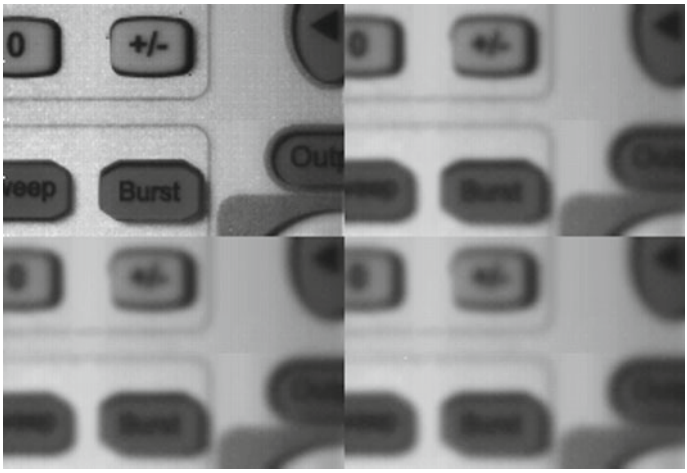

Fig. 20 Different snapshots of on-chip Gaussian pyramid

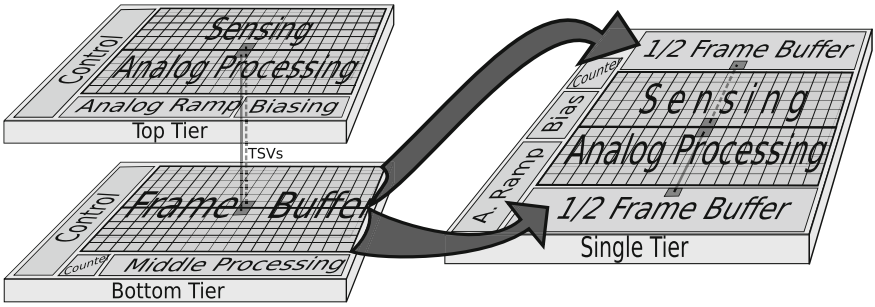

Fig. 21 Redistribution of circuits for Gaussian pyramid generation when mapping the original CMOS 3D-based architecture onto a conventional planar CMOS technology

5.3 Performance Comparison

Comparing the performance of the implemented prototypes with state-of-the-art focal-plane accelerator chips is not straightforward since every realization addresses a different functionality. As an example, we have included the most significant characteristics of our prototypes together with two recently reported focal-plane sensor-processor chips in Table 1. The Viola-Jones chip embeds extra functionalities in addition to the computation of the integral image [41] while featuring the largest resolution and the smallest pixel pitch, with a cost in terms of a reduced fill factor and increased energy consumption. Concerning the SIFT chip, one of the reasons of the energy overhead is the inherent high number of A/D conversions of the whole Gaussian pyramid plus the input scene, which amounts to 40 A/D conversions of the entire pixel array. Still, the acceleration at the focal plane provided by this chip

Table 1 Comparison of the implemented prototypes with state-of-the-art focal-plane sensor-processor chips

Reference	Ref. [42]	Ref. [43]	Viola-Jones chip	SIFT chip
Function	Edge filtering, tracking, HDR	2-D optic flow estimation	HDR, integral image, Gaussian filtering, programmable pixelation	Gaussian pyramid
Tech. (μm)	0.18	0.18	0.18	0.18
Supply (V)	0.5	3.3	1.8	1.8
Resolution	64 × 64	64 × 64	320 × 240	176 × 120
Pixel pitch (μm)	20	28.8	19.6	44
Fill factor (%)	32.4	18.32	5.4	10.25
Dyn. range (dB)	105	–	102	–
Power consumption (nW/px·frame)	1.25	0.89	23.9	26.5

pays off when comparing with more conventional solutions, as shown in Table 2. The power consumption of conventional CMOS imagers from Omnivision [44] featuring the image resolution tackled by the corresponding processor is incorporated in each of the entries related to conventional solutions. We have not accounted for accesses to external memories first because such costs would also be present if our chip were part of a complete hardware platform for a particular application; and

Table 2 Comparison of the SIFT focal-plane accelerator chip with conventional solutions

Hardware solution	Functionality	Energy/frame	Energy/pixel	Mpx/s
SIFT chip 180 nm CMOS	Gaussian pyramid	176 × 120 resol. 70 mW @ 8 ms 0.56 mJ/frame	26.5 nJ/px	2.64
Ref. [45] OV9655 + Core-i7	Gaussian pyramid	VGA resol. 90 mW @ 30 fps + 35 W @ 136 ms 4.8 J/frame	15.5 μJ/px	2.26
Ref. [46] OV9655 + Core-2-Duo	Gaussian pyramid	VGA resolution 90 mW + 35 W @ 2.1 s 73.7 J/frame	240 μJ/px	0.15
Ref. [47] OV6922 + Qualcomm Snapdragon S4	Gaussian pyramid	350 × 256 resol. 30 mW + 4 W @ 98.5 ms 0.4 J/frame	4.4 μJ/px	0.91

second because they are hardly predictable even with memory models. The energy cost of our chip outperforms that of an imager + conventional processor unit—even a low-power unit—in three orders of magnitude with similar processing speed. This leads to a combined speed-power figure of merit which makes our chip outperform conventional solutions in the range of three to six orders of magnitude.

6 Conclusions and Future Work

Focal-plane sensing-processing constitutes an architectural approach that can boost the performance of vision algorithms running on embedded systems. Specifically, early vision stages can greatly benefit from focal-plane acceleration by exploiting the distributed memory and concurrent processing in 2-D arrays of sensing-processing pixels. This chapter provides an overview of the fundamental concepts driving the design and implementation of two focal-plane accelerator chips tailored, respectively, for the Viola-Jones and the SIFT algorithms. These are the first steps within a long-term research framework aiming at achieving image sensors capable of simultaneously rendering high-resolution high-quality raw images and valuable pre-processing at ultra-low energy cost. The future work will be singularly biased by the availability of monolithic sensing-processing stacks. 3-D technologies will remove the tradeoff arising when it comes to allocating silicon area for sensors and processors on the same plane. High sensitivity and high processing parallelization will be compatible on the same chip. 3-D stacks will also foster alternative ways of making the most of vertical across-chip interconnections, from transistor level up to system architecture. In summary, 3-D integration technologies are the natural solution to develop feature extractors with low power budget without degrading image quality. Our prototypes on planar processes already consider future migration to these technologies, and this will continue to be a compulsory requirement of forthcoming designs.

Acknowledgments This work has been funded by: Spanish Government through projects TEC2012-38921-C02 MINECO (European Region Development Fund, ERDF/FEDER), IPT-2011-1625-430000 MINECO and IPC-20111009 CDTI (ERDF/FEDER); Junta de Andalucía through project TIC 2338-2013 CEICE; Xunta de Galicia through projects EM2013/038, AE CITIUS (CN2012/151, ERDF/FEDER), and GPC2013/040 ERDF/FEDER; Office of Naval Research (USA) through grant N000141410355.

References

1. Market analysis, embedded vision alliance. http://www.embedded-vision.com/industry-analysis/market-analysis
2. Kolsch, M., Butner, S.: Hardware considerations for embedded vision systems. In: Kisacanin, B., Bhattacharyya, S.S., Chai, S. (eds.) Embedded Computer Vision, Advances in Pattern Recognition Series, pp. 3–26. Springer, London (2009)
3. Open computing language. https://www.khronos.org/opencl/

4. OpenVX: portable, power-efficient vision processing. https://www.khronos.org/openvx/
5. Open source computer vision. http://opencv.org/
6. Compute unified device architecture. http://www.nvidia.com/object/cuda_home_new.html
7. Bailey, D.: Design for Embedded Image Processing on FPGAs. Wiley, Singapore (2011)
8. Kim, J., Rajkumar, R., Kato, S.: Towards adaptive GPU resource management for embedded real-time systems. ACM SIGBED Rev. **10**, 14–17 (2013)
9. Tusch, M.: Harnessing hardware accelerators to move from algorithms to embedded vision. In: Embedded Vision Summit. Embedded Vision Alliance, Boston (2012)
10. Horowitz, M.: Computing's energy problem (and what we can do about it). In: International Solid-State Circuits Conference (ISSCC), pp. 10–14. San Francisco (2014)
11. Wilkes, M.V.: The memory gap and the future of high performance memories. SIGARCH Comput. Archit. News **29**, 2–7 (2001)
12. Zarándy, A. (ed.): Focal-Plane Sensor-Processor Chips. Springer, New York (2011)
13. Campardo, G., Ripamonti, G., Micheloni, R.: Scanning the issue: 3-D integration technologies. Proc. IEEE **97**, 5–8 (2009)
14. Courtland, R.: ICs grow up. IEEE Spectr. **49**, 33–35 (2012)
15. Viola, P., Jones, M.: Robust real-time face detection. Int. J. Comput. Vis. **57**, 137–154 (2004)
16. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004)
17. Jia, H., Zhang, Y., Wang, W., Xu, J.: Accelerating Viola-Jones face detection algorithm on GPUs. In: IEEE International Conference on Embedded Software and Systems, pp. 396–403. Liverpool (2012)
18. Masek, J., Burget, R., Uher, V., Guney, S.: Speeding up Viola-Jones algorithm using multi-core GPU implementation. In: IEEE International Conference on Telecommunications and Signal Processing (TSP), pp. 808–812. Rome (2013)
19. Acasandrei, L., Barriga A.: FPGA implementation of an embedded face detection system based on LEON3. In: International Conference on Image Processing, Computer Vision, and Pattern Recognition. Las Vegas (2012)
20. Ouyang, P., Yin, S., Zhang, Y., Liu, L., Wei, S.: A fast integral image computing hardware architecture with high power and area efficiency. IEEE Trans. Circuits Syst. **II**(62), 75–79 (2015)
21. Kyrkou, C., Theocharides, T.: A flexible parallel hardware architecture for adaboost-based real-time object detection. IEEE Trans. Very Large Scale Integr. VLSI Syst. **19**, 1034–1047 (2011)
22. Gschwandtner, M., Uhl, A., Unterweger, A.: Speeding up object detection fast resizing in the integral image domain. Technical Report, University of Salzburg (2014)
23. de la Cruz, J.A.: Field-programmable gate array implementation of a scalable integral image architecture based on systolic arrays. Master Thesis, Utah State University (2011)
24. Kumar, G., Prasad, G., Mamatha, G.: Automatic object searching system based on real time SIFT algorithm. In: IEEE International Conference on Communication Control and Computing Technologies, pp. 617–622. Ramanathapuram (2010)
25. Cornelis, N., Van Gool, L.: Fast scale invariant feature detection and matching on programmable graphics hardware. In: IEEE Computer Vision and Pattern Recognition Workshops, pp. 1–8. Anchorage (2008)
26. Cohen, B., Byrne, J.: Inertial aided SIFT for time to collision estimation. In: IEEE International Conference on Robotics and Automation, pp. 1613–1614. Kobe (2009)
27. Cabani, C., MacLean, W.J.: A proposed pipelined-architecture for FPGA-based affine-invariant feature detectors. In: IEEE Computer Vision and Pattern Recognition Workshops, pp. 121. New York (2006)
28. Nobre, H., Kim, H.Y.: Automatic VHDL generation for solving rotation and scale-invariant template matching in FPGA. In: IEEE Southern Conference on Programmable Logic, pp. 21–26. Sao Carlos (2009)
29. Song, H., Xiao, H., He, W., Wen, F., Yuan, K.: A fast stereovision measurement algorithm based on SIFT keypoints for mobile robot. In: IEEE International Conference on Mechatronics and Automation (ICMA), pp. 1743–1748. Takamatsu (2013)

30. Gao, H., Yin, S., Ouyang, P., Liu, L., Wei, S.: Scale invariant feature transform algorithm based on a reconfigurable architecture system. In: 8th IEEE International Conference on Computing Technology and Information Management (ICCM), pp. 759–762. Seoul (2012)
31. Noguchi, H., Guangji H., Terachi, Y., Kamino, T., Kawaguchi, H., Yoshimoto, M.: Fast and low-memory-bandwidth architecture of SIFT descriptor generation with scalability on speed and accuracy for VGA video. In: IEEE International Conference on Field Programmable Logic and Applications (FPL), pp. 608–611. Milano (2010)
32. Andrea Vedaldi's implementation of the SIFT detector and descriptor. http://www.robots.ox.ac.uk/vedaldi/code/sift.html
33. Jahne, B.: Multiresolution signal representation. In: Jahne, B., Haubecker, H., Geibler, P. (eds.) Handbook of Computer Vision and Applications (volume 2). Academic Press, San Diego (1999)
34. Fernández-Berni, J., Carmona-Galán, R., del Río, R., Rodríguez-Vázquez, A.: Bottom-up performance analysis of focal-plane mixed-signal hardware for Viola-Jones early vision tasks. Int. J. Circuit Theory Appl. (2014). doi:10.1002/cta.1996
35. Fernández-Berni, J., Carmona-Galán, R., Carranza-González, L.: FLIP-Q: a QCIF resolution focal-plane array for low-power image processing. IEEE J. Solid-State Circuits **46**, 669–680 (2011)
36. Allen, P.E.: Switched Capacitor Circuits. Springer, New York (1984)
37. Suárez, M., Brea, V.M., Cabello, D., Pozas-Flores, F., Carmona-Galán, R., Rodríguez-Vázquez, A.: Switched-capacitor networks for scale-space generation. In: IEEE European Conference on Circuit Theory and Design (ECCTD), pp. 190–193. Linkoping (2011)
38. Enz, C.C., Temes, G.C.: Circuit techniques for reducing the effects of op-amp imperfections: autozeroing, correlated double sampling, and chopper stabilization. Proc. IEEE **84**, 1584–1614 (1996)
39. Suárez, M., Brea, V.M., Fernández-Berni, J., Carmona-Galán, R., Cabello, D., Rodríguez-Vázquez, A.: A 26.5 nJ/px 2.64 Mpx/s CMOS vision sensor for gaussian pyramid extraction. In: IEEE European Solid-State Circuits Conference (ESSCIRC), pp. 311–314. Venice (2014)
40. Suárez, M., Brea, V.M., Fernández-Berni, J., Carmona-Galán, R., Liñán, G., Cabello, D., Rodríguez-Vázquez, A.: CMOS-3-D smart imager architectures for feature detection. IEEE J. Emerg. Sel. Top. Circuits Syst. **2**, 723–736 (2012)
41. Fernández-Berni, J., Carmona-Galán, R., del Río, R., Kleihorst, R., Philips, W., R., Rodríguez-Vázquez, A.: Focal-plane sensing-processing: a power-efficient approach for the implementation of privacy-aware networked visual sensors. Sensors **14**, 15203–15226 (2014)
42. Yin, C., Hsieh, C.: A 0.5V 34.4μW 14.28kfps 105dB smart image sensor with array-level analog signal processing. In: IEEE Asian Solid-State Circuits Conference (ASSCC), pp. 97–100. Singapore (2013)
43. Park, S., Cho, J., Lee, K., Yoon, E.: 243.3pJ/pixel bio-inspired time-stamp-based 2D optic flow sensor for artificial compound eyes. In: IEEE International Solid-State Circuits Conference (ISSCC), pp. 126–127. San Francisco (2014)
44. Omnivision image sensors. http://www.ovt.com/products/
45. Murphy, M., Keutzer, K., Wang, P.: Image feature extraction for mobile processors. In: IEEE International Symposium on Workload Characterization (IISWC), pp. 138–147. Austin (2009)
46. Huang, F., Huang, S., Ker, J., Chen, Y.: High-performance SIFT hardware accelerator for real-time image feature extraction. IEEE Trans. Circuits Syst. Video Technol. **22**, 340–351 (2012)
47. Wang, G., Rister, B., Cavallaro, J.: Workload analysis and efficient openCL-based implementation of SIFT algorithm on a smartphone. In: IEEE Global Conference on Signal and Information Processing, pp. 759–762. Austin (2013)