Invited Paper

Focal-plane generation of multi-resolution and multi-scale image representation for low-power vision applications

J. Fernández-Berni[a], R. Carmona-Galán[a], L. Carranza-González[a],
A. Zarándy[b] and Á. Rodríguez-Vázquez[a]

[a]Institute of Microelectronics of Seville (IMSE-CNM), CSIC-University of Seville, Spain
[b]Computer and Automation Research Institute (MTA-SZTAKI), Budapest, Hungary

ABSTRACT

Early vision stages represent a considerably heavy computational load. A huge amount of data needs to be processed under strict timing and power requirements. Conventional architectures usually fail to adhere to the specifications in many application fields, especially when autonomous vision-enabled devices are to be implemented, like in lightweight UAVs, robotics or wireless sensor networks. A bioinspired architectural approach can be employed consisting of a hierarchical division of the processing chain, conveying the highest computational demand to the focal plane. There, distributed processing elements, concurrent with the photosensitive devices, influence the image capture and generate a pre-processed representation of the scene where only the information of interest for subsequent stages remains. These focal-plane operators are implemented by analog building blocks, which may individually be a little imprecise, but as a whole render the appropriate image processing very efficiently. As a proof of concept, we have developed a 176x144-pixel smart CMOS imager that delivers lighter but enriched representations of the scene. Each pixel of the array contains a photosensor and some switches and weighted paths allowing reconfigurable resolution and spatial filtering. An energy-based image representation is also supported. These functionalities greatly simplify the operation of the subsequent digital processor implementing the high level logic of the vision algorithm. The resulting figures, 5.6mW@30fps, permit the integration of the smart image sensor with a wireless interface module (Imote2 from Memsic Corp.) for the development of vision-enabled WSN applications.

Keywords: focal-plane image processing, reduced scene representation, power-efficient VLSI implementation, Wireless Sensor Networks

1. INTRODUCTION

Conventional processing architectures handle a purely digital signal flow until the outcome of the targeted result. All processing is carried out in the digital domain over data ultimately coming from an analog-to-digital conversion interface. When the data to be processed corresponds to 1-D signals, e.g. audio, this approach usually suffices to achieve a high throughput with low power consumption. However, for multi-dimensional signals, e.g an image sequence, the amount of information becomes so massive that conventional architectures fail to meet the specifications under strict timing and power requirements. On one hand, strict timing requirements can be only fulfilled by high-speed data processing, what in turn demands a digital processor running at high frequency. On the other hand, the dynamic power consumption of a digital processor is proportional to the frequency of its clock.[1] A tradeoff arises which is quite difficult to solve for applications requiring both conditions.

Nature gives us some hints on how to efficiently implement the processing of an image flow. In natural vision systems, the visual information is not only acquired but also pre-processed in the focal-plane device, the retina, before being sent to the visual cortex.[2] Interestingly, this pre-processing is performed in the analog domain by means of dedicated biological circuitry organized into layers.[3] The result is a retinotopic and simplified though elaborated version of the corresponding scene, i. e. less data but of a higher abstraction level. A clear example of the capability of this approach to extract only the relevant information from the visual stimulus is the human eye.

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

Infrared Technology and Applications XXXVII, edited by Bjørn F. Andresen, Gabor F. Fulop, Paul R. Norton,
Proc. of SPIE Vol. 8012, 80120E · © 2011 SPIE · CCC code: 0277-786X/11/$18 · doi: 10.1117/12.883881

In it, the information collected by about 150mill. photoreceptors is pre-processed by the retina and compressed into about 1mill. fibers composing the optic nerve.

Different prototype chips emulating the natural vision processing chain can be found in the literature.[4–6] These chips implement a massively parallel focal-plane array where each pixel does not consist only of a simple photosensor but also includes analog processing circuitry. The resulting pixel-level processor is usually 4- or 8-connected to its neighbors rendering a processing grid that makes use of the SIMD (Single Instruction Multiple Data) paradigm.[7] Thus, each element of the grid, that is, each pixel-level processor, executes the same instruction while operating over different data. This framework to pre-process images is especially suitable if we analyze the characteristics of low-level tasks,[8] commonly applied in early vision stages. To start with, low-level tasks feature a very regular computational flow, that is, all pixels are equally processed at every step. Therefore, few instructions applied to all pixels define the corresponding task. Additionally, the result of the computations associated with each pixel is usually independent from the result of the computations over the rest. This means that each pixel can be processed in parallel with the rest without distorting the outcome. And finally, a moderate accuracy (6-7 bits) suffices for this outcome in most cases. This enables the use of analog circuitry, not very precise but faster and more area- and power-efficient than its digital counterpart.

So far, the reported implementations based on the guidelines just described can be considered as general-purpose vision hardware capable of reaching excellent performance figures in terms of the ratio 'power consumption'/'computational power'. However, their power consumption as a whole makes them still too heavy for their incorporation to applications demanding really low power budgets. Bearing in mind this type of applications, we have designed a prototype vision chip called *FLIP-Q*, reported recently.[9] This chip also follows the guidelines above sketched but only a reduced subset of focal-plane processing primitives is implemented. These primitives deliver user-defined simplifications of the scene at ultra low energy cost. Indeed, the simplification of the scene is a key point for one of the application fields which can take significant advantage of the bioinspired approach for low-power image processing proposed: vision-enabled Wireless Sensor Networks (WSNs).[10]

All in all, in this paper we firstly review the processing capabilities of the *FLIP-Q* prototype and justify the choice of the focal-plane primitives implemented. We describe later the integration of *FLIP-Q* with a commercial WSN platform. Finally, some preliminary results extracted from the resulting system are presented.

2. *FLIP-Q*: POWER-EFFICIENT IMAGE PROCESSING FOR VISION-ENABLED AUTONOMOUS DEVICES

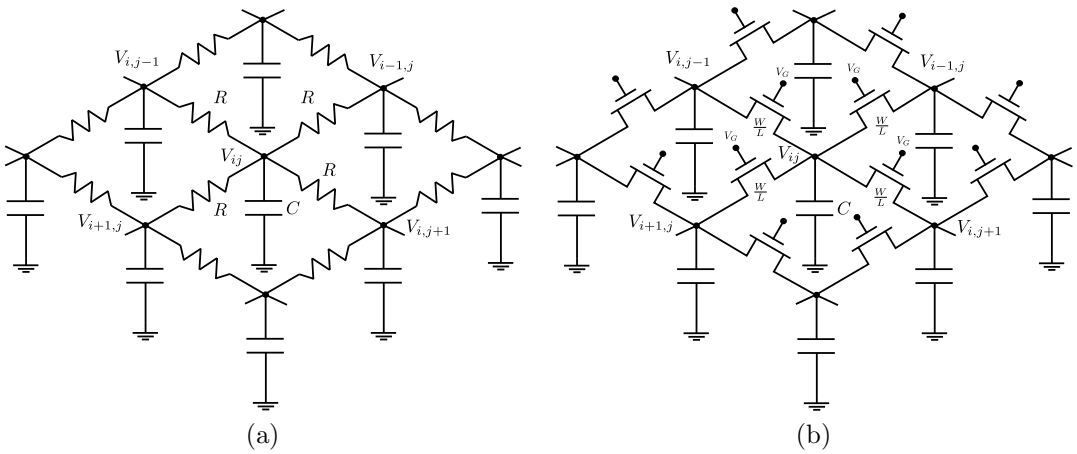
The efficient operation of *FLIP-Q* is mainly supported by the physical implementation of a diffusion process. The concept of diffusion is widely applied in physics. It explains the equalization process undergone by an initially uneven concentration of a certain magnitude. A typical example is heat diffusion. 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 PDE:

$$\frac{\partial V}{\partial t} = D\nabla^2 V \tag{1}$$

where D is referred to as the diffusion coefficient. We are assuming that D does not depend on the position and therefore an isotropic diffusion is taking place. After some transformations of Eq. (1), it is possible to demonstrate[11] that a diffusion process is equivalent to the convolution expressed by the following equation:

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

where $\sigma = \sqrt{2Dt}$. This equation shows that a diffusion process intrinsically entails a spatial Gaussian filtering varying 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. Another interesting property



(a) (b)

Figure 1. RC network performing isotropic diffusion (a) and its MOS-based counterpart (b).

of diffusion is that, for $t \to \infty$, only the dc component of $V(.)$ remains. Furthermore, this dc component is completely unaffected by the diffusion process itself.

Consider now the RC network depicted in Fig. 1(a). A real — spatially-discretized — diffusion process takes place within this circuit. An uneven charge distribution at the capacitors is diffused across the network and along time with a pace which is determined by the time constant $\tau = RC$. Eventually, a steady-state is reached when the charge is evenly distributed. Note that no additional energy is necessary for the network to evolve apart from the initial charging of the nodes. This means, taking into account the linear relation between charge and voltage in a capacitor, that if we map the pixel values of an image into the initial voltages at the capacitors, a family of Gaussian filters can be applied to such an image. And this can be done without energy cost by simply letting the network to evolve. Two problems mainly arise regarding the VLSI implementation of this circuit. First of all, it is necessary to stop the dynamics of the network at user-defined time instants in order to obtain targeted Gaussian filters. Simple resistors linking the nodes can not be used to this end as they only have 2 terminals. Secondly, the low sheet resistance exhibited by the most resistive materials available in standard CMOS requires very large areas for the necessary values of resistance. We have demonstrated[12] that these two problems can be solved by the MOS-based counterpart depicted in Fig. 1(b). The use of MOS transistors biased in the ohmic region instead of resistors enables the control of the network dynamics through the gate terminals. Besides, their ratio resistance/area is much greater than that of the resistors made with polysilicon or diffusion strips. As a result, and despite the unavoidable nonlinearities of the transistors, equivalent resolutions around 6-7 bits are obtained from the MOS-based RC network implemented in the *FLIP-Q* prototype.

By combining the programmable filtering delivered by a time-controlled MOS-based RC network with reconfigurable block-wise image plane division, the image processing capabilities are boosted. Thus, in *FLIP-Q*, it is possible to extract information about different spatial frequency bands at user-defined regions of the image plane. Also the possibility of computing the dc component of a group of pixels by means of a long enough diffusion allows for multi-resolution and foveated scene representations. And, even more importantly, the axioms of linearity, shift invariance, semi-group structure, and not enhancement of local extrema held by the Gaussian kernel associated with the diffusion process permit the generation of independent scale spaces[13] in sub-divisions of the focal plane. An example of this operation can be seen in Fig. 2. Scale spaces constitute a framework for image processing[14] that makes use of the representation of a scene at multiple scales. It is useful for example to detect scale-invariant features that characterize a scene.[15]

The last primitive implemented in *FLIP-Q* is also based on the diffusion carried out by the MOS-based RC network as well as on the reconfigurability of the focal-plane. Each pixel-level processor comprising the focal-plane array includes the simple circuit depicted in Fig. 3. The voltage $V_{ij}(t)$ represents the value of the corresponding pixel after performing diffusion for t seconds and stopping the network dynamics. Then, by using the transistor M_E and the switch S_E, the energy associated with the pixel at that point of the diffusion process

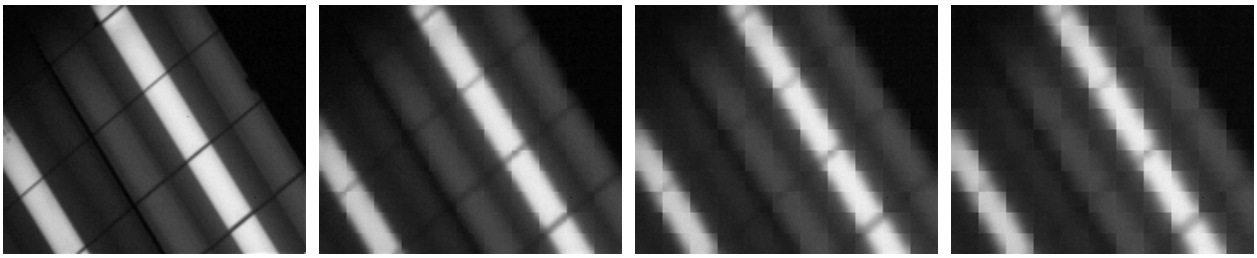


Figure 2. Independent scale spaces within focal-plane sub-divisions of 16×12px.

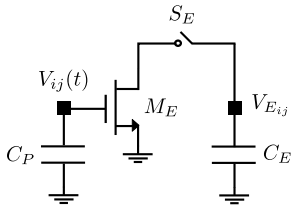


Figure 3. In-pixel circuit for the computation of the energy.

can be computed at $V_{E_{ij}}$, that is:

$$V_{E_{ij}} = k|V_{ij}(t)|^2 \tag{3}$$

where k is, ideally, a constant whose value depends on several technological parameters. Finally, thanks to the interconnection of each pixel-level processor with its neighborhood, the energy of a set of pixels can be attained. Thus, considering a regular focal-plane division where each block is composed of $W \times H$ pixels, the energy of the block (k, l) computed by the hardware will be:

$$E_{kl}(t) = k' \sum_{i=1}^{W} \sum_{j=1}^{H} |V_{ij_{kl}}(t)|^2 \tag{4}$$

where k' is again a constant which depends on technological parameters. The point is that this block-wise energy summarizes the diffusion realized independently within eack block in only one value. The longer the diffusion interval t the less $E_{kl}(t)$. The energy lost between two time instants during the difussion corresponds to that of the spatial frequencies filtered whereas the energy remaining at the end of the diffusion is associated exclusively with the dc component. Consequently, $E_{kl}(t)$ can be used as an indicator of the frequency content of the block (k, l). This energy-based representation can be used for example to estimate the salient regions in a scene. The difference between the initial value of the energy and the energy after a long enough diffusion accounts for the contrast within the block considered. The more the value of this difference, the larger the intensity changes which determine the frequency content of the block. An example of this operation, computing the difference of energy values off the chip, is shown in Fig. 4. Each block has a size of 4×4px.

We can see that all the processing primitives implemented in *FLIP-Q* are oriented to deliver a programmable simplification of the scene. The objective is that the image sensor enabling vision in a low-power device becomes a smart peripheral capable of adapting to the requirements of the running algorithm. The capture of each image composing a sequence will be determined by the characteristics of the objects to be analyzed at the moment. Thus, the image sensor will not output raw but pre-processed images that make the subsequent digital processing much lighter. The point is that the energy cost of such pre-processing must be lower than the energy cost of directly processing the raw representation of the scene. In the case of *FLIP-Q*, the maximum power consumption measured for the capture, processing and A/D conversion of an image flow at 30fps, with full-frame processing but reduced frame size output, is 5.6mW. We will see shortly that this figure represents less than 5% of the whole system power consumption for a vision-enabled WSN node.

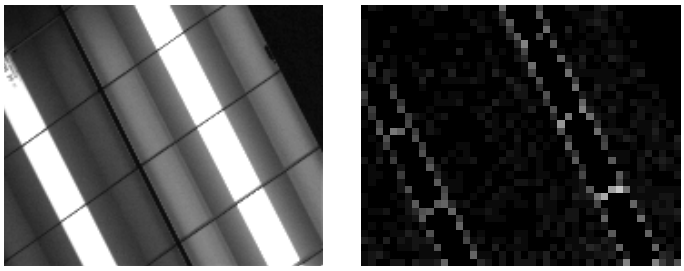


Figure 4. Example of salient region estimation based on the block-wise energy computation.

3. *WI-FLIP*: A WIRELESS FOCAL-PLANE LOW-POWER IMAGE PROCESSOR

Wi-FLIP is the system resulting from the integration of *FLIP-Q* and *Imote2*, a commercial WSN platform from MEMSIC Corp. This platform (see Fig. 5) is built around the 32-bit ARM5 Marvell PXA271 XScale® processor running *TinyOS*.[16] The PXA271 processor, which can operate in a low voltage (0.85V), low frequency (13MHz) mode, hence enabling very low power operation, is really a multi-chip module including 256kB SRAM, 32MB SDRAM and 32MB of FLASH memory. An 802.15.4-compliant radio is integrated into the *Imote2* system too. To supply the processor with all the required voltage domains, a Power Management Integrated Circuit (PMIC) is included. This PMIC supplies 9 voltage domains to the processor in addition to the dynamic voltage scaling capability. It also includes a battery charging option and battery voltage monitoring. *Imote2* was designed to support primary and rechargeable batteries through an attachable battery board as well as to being powered via USB. Note that external sensor boards can be connected through expansion connectors. We have used these connectors to interconnect the *Imote2* platform with the *FLIP-Q* prototype. The interconnection has been carefully designed according to the number of PXA271's GPIOs available. Specifically, there are 34 GPIOs which can be accessed through the 40-pin connector of the "advanced sensor board" interface of *Imote2*. Only the strictly necessary logic to configure the processing primitives implemented by the *FLIP-Q* sensor and retrieve the corresponding outcome is mapped into these GPIOs. Those signals included in the prototype for test purposes are dismissed. In order to implement this interconnection plan and supply power and biasing to the prototype, a 2-layer PCB has been designed and fabricated. Two snapshots of the resulting vision-enabled WSN node are shown in Fig. 6.

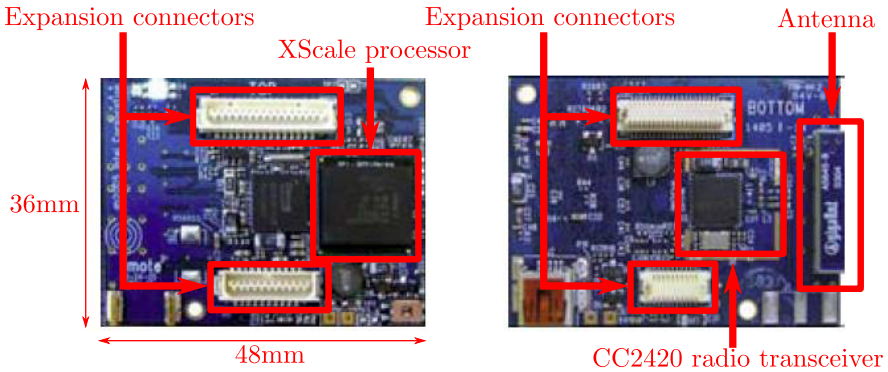


Figure 5. Top view (left) and bottom view (right) of the *Imote2* platform.

4. PRELIMINARY TESTS

The first step to realize any experimental test with the *Wi-FLIP* platform is to write the corresponding program, compile it and download it into the PXA271 processor. The standard programming language to develop applications running on *TinyOS* is *nesC* (network embedded system C).[17] We have used the widely known *cygwin* environment to cross-compile *nesC* code for the PXA271 processor. The resulting native code, ready to be

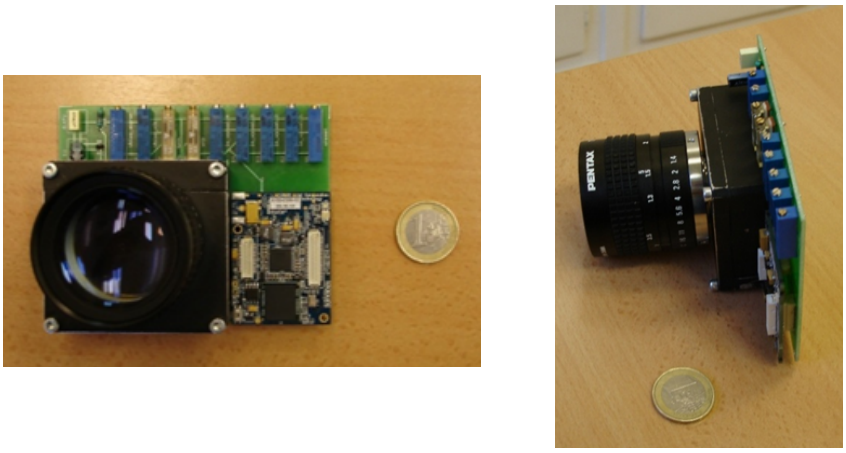


Figure 6. *Wi-FLIP*: a vision-enabled node for wireless applications.

executed, is downloaded into the mote via USB. In this first version of *Wi-FLIP*, the biasing signals for *FLIP-Q* must be manually adjusted through potentiometers before executing any code.

An operation typically needed for artificial vision applications is edge detection. This operation can be realized through Difference of Gaussians (DoG).[18] In our case, the difference between a non-filtered image and a Gaussian-filtered version of that same image will be computed. We can afford this simplification because of the low noise associated to the frames captured by *FLIP-Q*, what enables the possibility of skipping the application of a first Gaussian filter to eliminate high-frequency noise. We make use of the MOS-based RC network to apply the only Gaussian filter needed. The absolute value of the pixel difference between the original non-filtered image and the filtered image is calculated at the PXA271 processor. Two original images and their corresponding edge filtered version directly downloaded from *Wi-FLIP* are depicted in Fig. 7. The algorithm developed firstly performs an adaptation of the exposure time, T_{exp}, to the characteristics of the scene at the moment. Operating in photocurrent integration mode, the voltage V_{ij} representing the value of each pixel depends on T_{exp}. Thus, for the same power of incident light over the sensor surface, a larger or smaller value of T_{exp} will result respectively in a larger or smaller excursion of V_{ij} from the reset voltage. If T_{exp} is not correctly set, we will obtain too dark or too bright images. A simple mechanism to adjust T_{exp} is to force that the mean value of the image falls around the middle point of the nominal pixel voltage range. In this way, we make sure that most of the pixels are neither over-exposed nor under-exposed according to the current conditions of the scene. We also use the MOS-based RC network to compute the mean value of the image by realizing charge redistribution concurrently with photointegration.

Currently, the only drawback for the implementation of vision algorithms in *Wi-FLIP* is the low frame rate reachable. In the case of the edge detection algorithm, the maximum frame rate achieved for full-resolution images is 0.1fps. This figure is obtained by setting the frequency of the PXA271's clock to 416MHz, what means a power consumption of around 600mW. For the minimum possible clock frequency, 13 MHz, the frame rate is 0.01fps with a power consumption of around 150mW. The bottleneck preventing *Wi-FLIP* from achieving higher frame rates is the control of the A/D conversion at *FLIP-Q* by *Imote2*. This control, that is not standard and must be therefore programmed step by step in *nesC*, is mostly supported by GPIOs featuring very slow switching. Furthermore, the software overhead introduced by *TinyOS* also plays an important role. As a consequence, a great deal of clock cycles is wasted during the conversion. For instance, only the frame conversion for full, half and quarter resolution at maximum clock speed, i. e. 416MHz, takes respectively 3.9s, 1s and 0.3s. It is therefore mandatory for future versions of *FLIP-Q* either the incorporation of internal digital logic realizing efficiently the ADC control or the implementation of a standard interface that speeds up this task, like for example the Quick Capture Interface provided by the PXA271 processor.

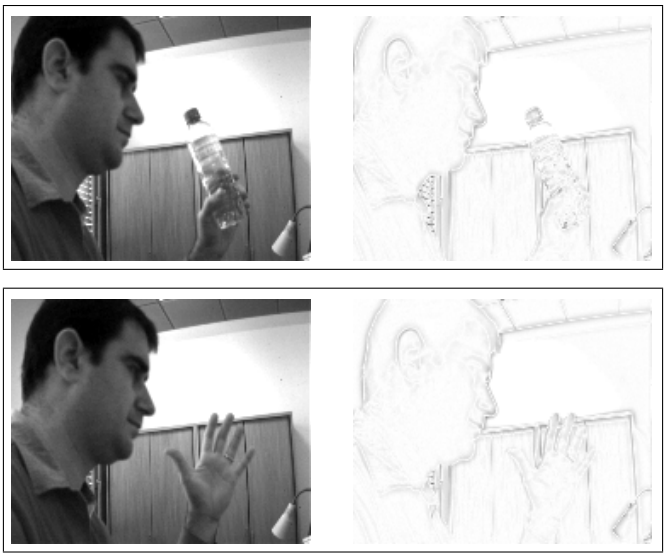


Figure 7. Two frames after running the edge detection algorithm in *Wi-FLIP*.

5. CONCLUSIONS

Early vision tasks feature a regular computational flow that can be applied in parallel to the pixels composing an image. Besides, they do not demand a very accurate operation. Massively parallel analog focal-plane arrays implementing a SIMD-based processing architecture adapt very well to these properties. They reach a high performance when carrying out low-level image processing, making subsequent digital stages much lighter and consequently improving the performance of the whole vision system. We have demonstrated these claims with a prototype vision chip designed ad-hoc for low-power applications. This prototype has been incorporated as a peripheral into a commercial WSN platform barely increasing its power consumption. Currently, the only problem is the non-standard interface through which the interconnection is realized. It forces the use of GPIO ports making to retrieve data from the prototype extremely slow. The implementation of a standard interface would speed up greatly this operation, improving in turn the throughput of the system.

ACKNOWLEDGMENTS

This work is partially funded by the Andalusian regional government (Junta de Andalucía-CICE) through project 2006-TIC-2352 and the Spanish Ministry of Science (MICINN) through project TEC 2009-11812, co-funded by the European Regional Development Fund, and also supported by the Office of Naval Research (USA), through grant N000141110312.

REFERENCES

[1] Rabaey, J., [*Digital Integrated Circuits: A Design Perspective*], Prentice Hall (1995).

[2] Masland, R., "The fundamental plan of the retina," *Nature Neurosci.* **4**(9), 877–886 (2001).

[3] Roska, B. and Werblin, F., "Vertical interactions across ten parallel, stacked representations in the mammalian retina," *Nature* **410**, 583–587 (2001).

[4] Linan-Cembrano, G., Rodriguez-Vazquez, A., Carmona-Galan, R., Jimenez-Garrido, F., Espejo, S., and Dominguez-Castro, R., "A 1000 FPS at 128x128 vision processor with 8-bit digitized I/O," *IEEE J. of Solid-State Circuits* **39**(7), 1044–1055 (2004).

[5] Dudek, P. and Hicks, P., "A general-purpose processor-per-pixel analog SIMD vision chip," *IEEE Trans. Circuits Syst. I* **52**(1), 13–20 (2005).

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

[7] Unger, S., "A computer oriented toward spatial problems," *Proceedings of the IRE* **46**(10), 1744–1750 (1958).
[8] Gonzalez, R. and Woods, R., [*Digital Image Processing*], Prentice Hall (2002).
[9] 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. of Solid-State Circuits* **46**(3), 669–680 (2011).
[10] Akyildiz, I., Melodia, T., and Chowdhury, K., "A survey on wireless multimedia sensor networks," *Computer Networks* **51**(4), 921–960 (2007).
[11] Jahne, B., [*Handbook of Computer Vision and Applications (volume 2)*], ch. 4, 67–90, Academic Press (1999).
[12] 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* (2011). DOI 10.1002/cta.564.
[13] Babaud, J., Witkin, A. P., Baudin, M., and Duda, R. O., "Uniqueness of the gaussian kernel for scale-space filtering," *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 26–33 (1986).
[14] Lindeberg, T., "Feature detection with automatic scale selection," *International Journal of Computer Vision* **30**(2), 79–116 (1998).
[15] Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Int. J. of Computer Vision* **60**(2), 91–110 (2004).
[16] Levis, P. and Gay, D., [*TinyOS Programming*], Cambridge University Press, New York, NY (USA) (2009).
[17] Gay, D., Levis, P., von Behren, R., Welsh, M., Brewer, E., and Culler, D., "The nesc language: A holistic approach to networked embedded systems," in [*Proc. of Conf. on Programming Language Design and Implementation (PLDI)*], 1–11 (2003).
[18] Poggio, T., Voorhees, H., and Yuille, A., "A regularized solution to edge detection," *J. of Complexity* **4**(2), 106–123 (1988).