

# Ultra-low-power processing array for image enhancement and edge detection

Jorge Fernández-Berni, Ricardo Carmona-Galán, *Associate Member, IEEE* and  
Ángel Rodríguez-Vázquez, *Fellow, IEEE*

**Abstract**—This paper presents a massively parallel processing array designed for the 0.13 $\mu\text{m}$  1.5V standard CMOS base process of a commercial 3-D TSV stack. The array, which will constitute one of the fundamental blocks of a smart CMOS imager currently under design, implements isotropic Gaussian filtering by means of a MOS-based RC network. Alternatively, this filtering can be turned into anisotropic by a very simple voltage comparator between neighboring nodes whose output controls the gate of the elementary MOS resistor. Anisotropic diffusion enables image enhancement by removing noise and small local variations while preserving edges. A binary edge image can be also attained by combining the output of the voltage comparators. In addition to these processing capabilities, the simulations have confirmed the robustness of the array against process variations and mismatch. The power consumption extrapolated for a VGA-resolution array processing images at 30fps is 570 $\mu\text{W}$ .

**Index Terms**—Diffusion, RC network, Gaussian filtering, anisotropic filtering, resistive fuse, low-power, voltage comparator

## I. INTRODUCTION

3-D IC technologies [1], [2] are changing the way in which circuit design has been traditionally approached. The development of new techniques, from transistor level up to system architecture, which take advantage of vertical across-chip interconnections will dramatically boost the performance of the targeted functionality. In particular, with regard to smart CMOS imagers based on focal-plane sensing-processing [3], the availability of TSVs removes the trade-off arising when it comes to allocate silicon area for sensors and processors on the same plane. A top sensor layer can be now integrated and vertically interconnected with other layers exclusively dedicated to processing. Consequently, smart imagers with fill factors close to 100% and very high resolution can be achieved. When compared to planar implementations [4], [5], the absence of photosensitive devices at the processing layers releases a significant amount of area to be occupied by processing circuitry. However, this circuitry represents a major source of power consumption, especially for early vision tasks, where

the lattice of processing elements must ideally have a similar resolution to that of the sensor layer. The design of ultra-low-power building blocks is therefore mandatory to exploit the additional computational power provided by the vertical integration without shooting up the power consumption.

This paper focuses on this crucial issue. We present a massively parallel processing array for image enhancement and edge detection designed for the 0.13 $\mu\text{m}$  1.5V standard CMOS base process of the TSV stack commercialized by Tezzaron Semiconductor. The backbone of the array is a MOS-based RC network carrying out isotropic diffusion [6]. On this, we incorporate now a low-power time-controlled voltage comparator between neighboring nodes. The output of each comparator enables diffusion only between those nodes whose difference is less than a programmable threshold. Anisotropic filtering is thus performed, preserving large pixel differences — that is, edges — while suppressing noise and small local variations. Additionally, the outputs of the comparators are combined in order to deliver a binary edge image. The array features no static consumption apart from leakage currents. A dynamic consumption of 570 $\mu\text{W}$  for VGA resolution at 30fps is extrapolated from simulation. Finally, the proposed circuitry, which will be part of a smart CMOS imager providing additional early vision capabilities, can be reprogrammed to accommodate global process variations, being also robust against mismatch.

## II. RELATED WORK

The primary drawback of multiscale image description based on isotropic diffusion was clearly spelled out in [7]. Gaussian filtering does not distinguish between natural boundaries of objects and essentially flat regions containing only noise or textures. As a result, the edges are shifted as coarser scales are generated, preventing them from being accurately located. In order to solve this issue, a new definition of the scale-space representation is proposed [7]. It is also based on the diffusion equation, but the diffusion coefficient is now chosen to vary spatially at any scale, that is:

$$\frac{\partial V(x, y, t)}{\partial t} = \nabla \cdot [D(x, y, t)\nabla V(x, y, t)] \quad (1)$$

where  $V(x, y, t)$  is a brightness function defined over a continuous plane and  $D(x, y, t)$  represents the spatially-variant diffusion coefficient. The point is tuning this coefficient across the plane in such a way that intraregion filtering is given priority over filtering across region boundaries.

Manuscript received ???; revised ???. Copyright (c) 2012 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org This work is funded by MICINN (Spain) through projects TEC2009-11812 and IPT-2011-1625-430000, co-funded by the European Regional Development Fund, by the Office of Naval Research (USA) through grant N000141110312 and by the Spanish Centre for Industrial Technological Development, co-funded by the European Regional Development Fund, through Project IPC-20111009.

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

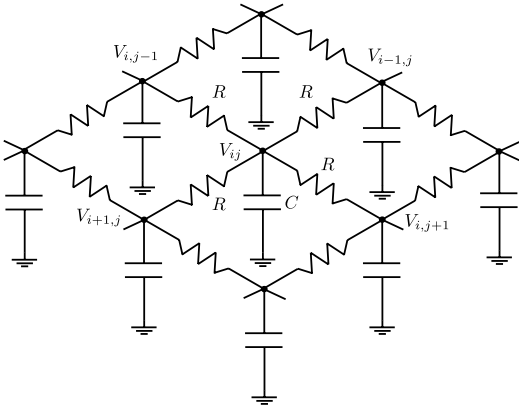


Fig. 1. RC network performing linear diffusion.

Resistive networks [8] constitute the basis for most of the VLSI implementations of this content-aware multiscale representation. These networks render a distribution of either currents or voltages that is equivalent to apply a diffusion process during a certain time interval over the input sources. In order to achieve anisotropic filtering, resistive fuses are introduced. These circuit elements behave as resistances of value  $R$  only when the voltage difference across their terminals is less than a certain threshold  $V_{\text{off}}$ . Otherwise, they behave as open circuits. By adjusting  $V_{\text{off}}$ , edges are prevented from being filtered while small brightness variations do undergo smoothing. A spatially-variant diffusion coefficient is thus emulated. There are however two significant problems associated with the first practical implementations of resistive fuses reported [9]–[11]. To start with, they rely on subthreshold operation. This makes them very dependent of the characteristics of the process, parameter variations and mismatch. And second, this dependence causes in turn a very restricted range for  $R$  and  $V_{\text{off}}$ , greatly constraining the set of filters attainable. These aspects are solved differently in [12] and [13]. Discrete-time switched capacitor techniques are applied in [12] in order to implement the horizontal resistors of a resistive network. The control of the amount of charge exchanged between neighboring nodes enables tunnable filtering.  $V_{\text{off}}$  also features a wider voltage swing. In [13], the image pixels are represented by means of currents. The current difference between neighboring nodes is compared to a programmable threshold. The binary output of this comparison controls the gate of a single transistor acting as the horizontal resistor of a resistive network. The amount of smoothing realized can be adjusted through the common-mode input level of the pixels.

Despite these successful implementations in terms of functionality, resistive networks present a major drawback: their static consumption. The input sources must continuously inject current into the grid in order to get the filtering done. Alternatively, Gaussian filtering can be also achieved by an RC network like that of Fig. 1. A real diffusion process takes place now within the network. 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  $\tau = RC$ . We demonstrated in [6] that an accurate approximation of an

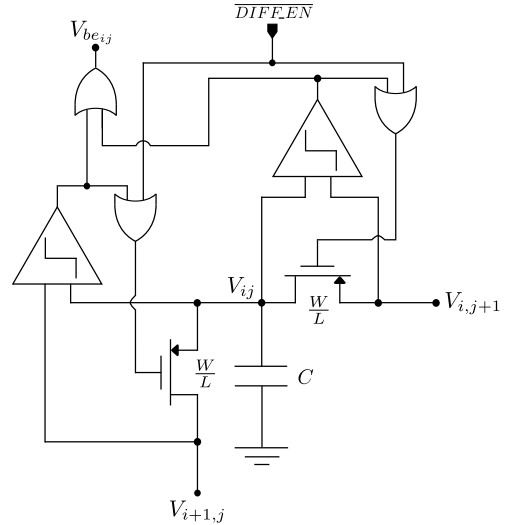


Fig. 2. Elementary cell of the anisotropic diffusion network proposed.

ideal RC network can be obtained by substituting every resistor by a MOS transistor biased in the ohmic region. Moreover, the value of the pixels can be easily mapped to the initial conditions of the capacitors and, without any additional energy contribution, the network will carry out isotropic filtering. In order to turn this filtering into anisotropic, the MOS resistors must be turned into resistive fuses. And to accomplish it, we propose a time-controlled voltage comparator based on a differential pair. The inputs of this comparator correspond to the neighboring nodes linked by a MOS resistor. Its binary output, together with a global diffusion enable signal, determines whether that resistor is activated or turned off. A time-controlled elementary resistive fuse is thus emulated. We must remark at this point that the MOS-based RC network for anisotropic filtering proposed in [13] works quite differently. It makes intensive use of current mode operation, leading again to high static consumption.

### III. ELEMENTARY CELL

Consider Fig. 2. It corresponds to the elementary cell of the network in Fig. 1 after substituting the elementary resistors interconnecting neighboring cells by MOS transistors biased in the ohmic region and incorporating additional circuitry to achieve anisotropic diffusion. Notice that the south and east connectivity suffice to make up the entire grid. Cells located at the bottom and rightmost edges will not obviously include south and east connection, respectively. The key component of this elementary cell is the voltage comparator. It outputs a logic ‘1’ when the absolute value of the difference between its input voltages exceeds a certain programmable threshold  $V_{\text{off}}$ . This turns the corresponding transistor off, no matter the logic value of the global active-low diffusion enable signal  $\overline{DIFF\_EN}$ . If the output of the comparator is ‘0’, diffusion between the neighboring nodes will take place during the time interval  $t$  in which  $\overline{DIFF\_EN}$  is active, that is, is set to ‘0’. The output of the comparators is also combined in order to obtain a binary edge image represented by  $V_{beij}$ .

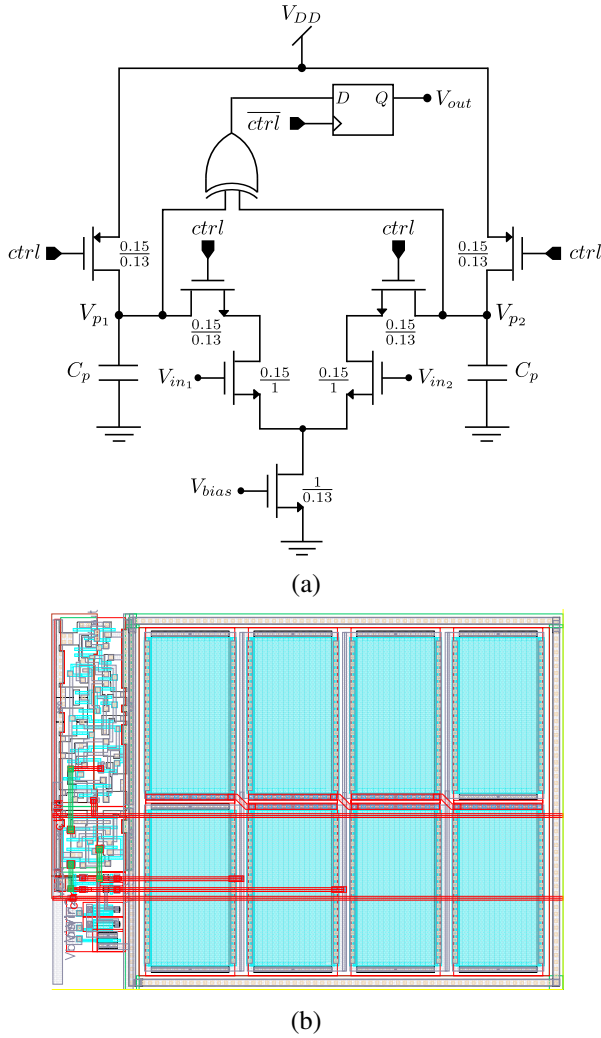


Fig. 3. Time-controlled voltage comparator (a) and its layout (b).

We propose to implement the voltage comparator as depicted in Fig. 3(a). It is based on a differential pair where the current-to-voltage conversion is not carried out as usual, that is, with resistive or MOS loads. In order to reduce the power consumption, the conversion takes place on the capacitors  $C_p$ . They are first precharged by turning on the pMOS switches connected to  $V_{DD}$  for  $T_p$  seconds. Subsequently, they are discharged during a certain time interval  $T_d$ . The pace of discharge for each capacitor is determined by the current flowing through each branch of the differential pair. Small differences between  $V_{in1}$  and  $V_{in2}$  will imply small differences between  $V_{p1}$  and  $V_{p2}$  at the end of the discharge whereas remarkably different input voltages will cause large differences between  $V_{p1}$  and  $V_{p2}$ . By adequately adjusting  $T_d$  through the control signal  $ctrl$ , both  $V_{p1}$  and  $V_{p2}$  can be situated over the input threshold voltage of the XOR gate for differences between the input voltages less than  $V_{Off}$ . In such a case, the output of the XOR is set to ‘0’. For differences larger than  $V_{Off}$ , either  $V_{p1}$  or  $V_{p2}$  will be situated over the gate input threshold whereas the other one will be situated below, setting the output to ‘1’. Notice that this output is latched in order to avoid the effects of leakage over  $V_{p1}$  and  $V_{p2}$ .

Let us analyze how  $V_{Off}$  can be adjusted. Consider a signal range of  $[0.75V, 1.5V]$  for the pixel values. This range is chosen according to the criteria described in [6]. Bear in mind that the pixels are represented by the voltages  $V_{ij}$ , which in turn constitute the inputs of the voltage comparators across the network. Suppose a voltage comparator where  $V_{in1} = 0.75V$  and  $V_{in2} = 0.8V$ . The corresponding voltage difference,  $0.05V$ , is considered too small to belong to an edge. Consequently, the comparator must output a logic ‘0’. However, for the same  $V_{in1}$  but  $V_{in2} = 1V$ , the difference reaches  $0.25V$ . The existence of an edge is assumed now and the comparator must therefore output a logic ‘1’. We have depicted in Fig. 4 the precharge of the capacitors to  $V_{DD}$  for  $30ns$  and the subsequent discharge for different time intervals,  $40ns$ ,  $70ns$  and  $120ns$ , in the two cases above proposed. In these simulations,  $V_{DD}$  is  $1.5V$ ,  $C_p$  is  $200fF$  and  $V_{bias}$  is  $0.6V$ , what is translated into a bias current of  $3.5\mu A$ . We have used the transistor models in HSPICE provided by Tezzaron/Globalfoundries as well as standard cells of the technology. For the sake of reference, the straight line crossing each diagram corresponds to the input threshold voltage of the standard XOR gate. It can be seen that, for  $V_{in1} = 0.75V$  and  $V_{in2} = 0.8V$ , intervals of  $T_d = 40ns$  and  $T_d = 70ns$  still keep  $V_{p1}$  and  $V_{p2}$  over the gate input threshold, as targeted. However, an interval of  $T_d = 120ns$  is too long since  $V_{p2}$  reaches a voltage below that threshold, setting the output to ‘1’. For  $V_{in1} = 0.75V$  and  $V_{in2} = 1V$ , a discharge of  $40ns$  is too short to highlight the pixel difference from the point of view of the XOR gate. As a result, the output is set to ‘0’. The adequate output is attained for discharges of  $70ns$  and  $120ns$ . It can be therefore concluded that  $T_d = 70ns$  achieves the targeted behavior for both cases. Notice that, because of the intrinsic structure of the comparator, any  $V_{in2}$  greater than  $1V$  — keeping  $V_{in1} = 0.75V$  and  $T_d = 70ns$  — will make  $V_{p2}$  go below the gate input threshold. As  $V_{p1}$  stays, under these conditions, always above it, no matter the value of  $V_{in2}$  considered, any pixel difference greater than  $0.25V$  will be processed as an edge. In order to demonstrate the programmability of  $V_{Off}$ , we have set  $V_{in1} = 0.75V$  and then  $V_{in2}$  has been swept from  $0.8V$  up to  $1.5V$  with steps of  $0.05V$ . This sweeping has been simulated for different intervals of discharge, registering for each the smallest pixel difference from which the voltage comparator starts to output a logic ‘1’. In other words, we have obtained the voltage  $V_{Off}$  featured by the elementary MOS resistive fuse of the RC network for each interval. The result is that  $V_{Off}$  ranges from  $0.6V$  to  $0.05V$  for discharges between  $50ns$  and  $120ns$ , respectively. The effect of the parasitics over these intervals is negligible, according to the simulations of the extracted layout shown in Fig. 3(b). Due to the nonlinearity of the transistors, these values of  $V_{Off}$  will undergo distortion when considering other possible combinations of input voltages, that is, other possible sweepings. However, this distortion can be compensated via calibration of  $V_{Off}$ , as described in Section IV.

Finally, a comment about the power consumption and area usage of the elementary cell just described. The efficiency of the MOS-based RC network was previously mentioned and experimentally proved in [6]. The energy cost of the digital cells

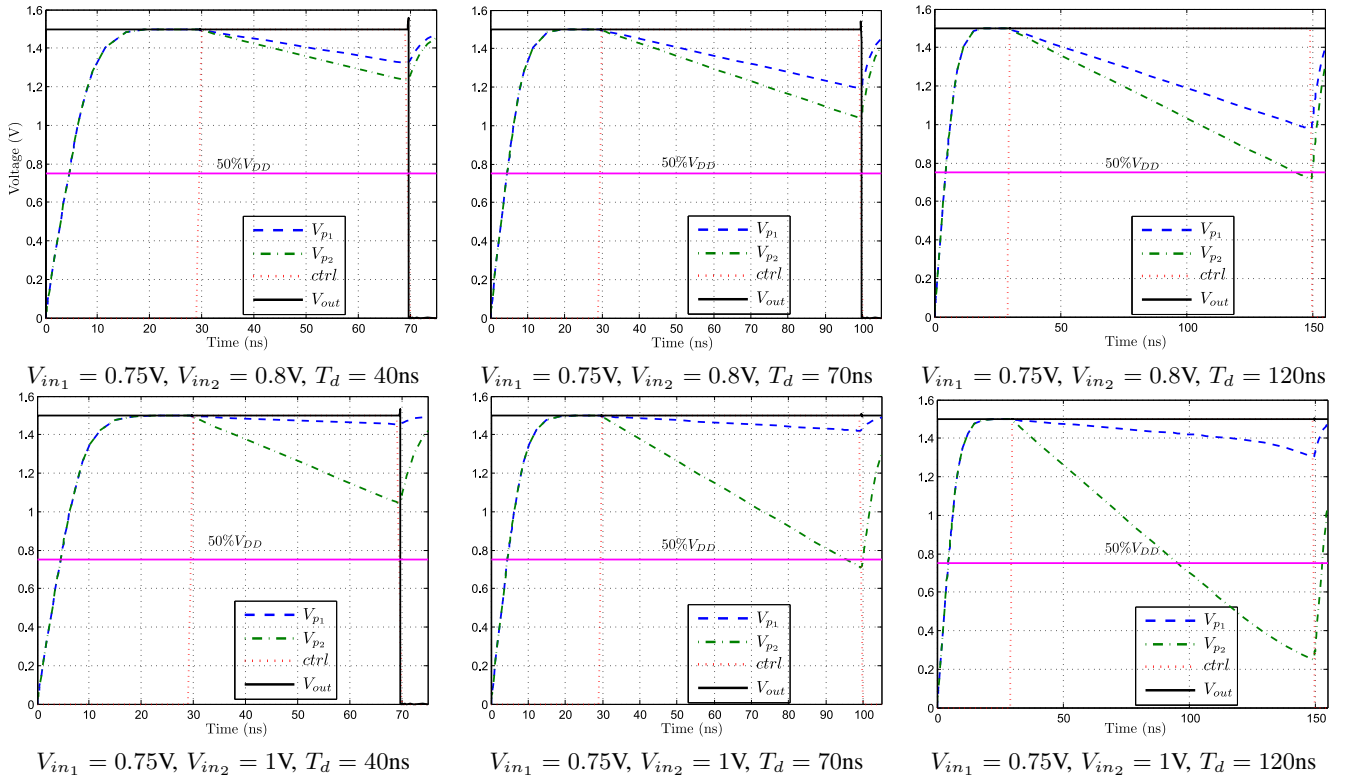


Fig. 4. Simulation results of the voltage comparator for different input voltages and different time intervals of discharge.

included now, with an order of magnitude of tens of nW/MHz at most, is also really small for typical frame rates of most vision applications. The main source of power consumption is the precharge of the capacitors at the voltage comparators. It must be noticed that the bias current simply makes the charge stored at these capacitors flow through the differential pair. No further energy contribution is involved in its operation. All in all, according to the simulations realized and the datasheet of the standard cells used, a pessimistic estimation of power consumption would be 1.85nW per elementary cell at 30fps. Extrapolating this figure to an VGA array, the total power consumption would be 570 $\mu$ W at 30fps. This calculation exclusively considers the power consumption associated with the processing capabilities of the array. We have therefore obviated the energy cost of mapping the pixel values to the initial conditions of the capacitors in the RC network. Concerning the area usage, we must say that the capacitors finally implemented in the targeted CMOS smart imager are going to be smaller than those considered in this paper. These elements, with their current values, require the largest area by far. Thus, for example, in the layout of Fig. 3(b), each capacitor of 200fF has been realized by using four MOS capacitors of dimensions 4.06 $\times$ 8.14 $\mu$ m<sup>2</sup>, leading to a total comparator area of 19.72 $\times$ 16.62 $\mu$ m<sup>2</sup>. Smaller capacitors, apart from possible mismatch considerations, simply mean changes in the timing of the control signals, making the processing dynamics faster.

#### IV. SIMULATION OF A 32 $\times$ 32 ARRAY

In order to corroborate the results obtained for a single elementary cell, we have built a 32 $\times$ 32 array in HSPICE.

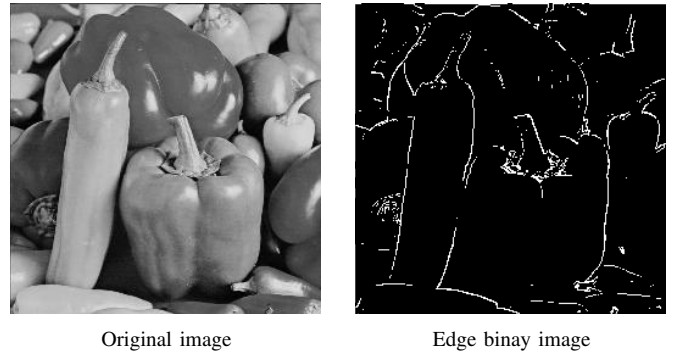


Fig. 5. Simulation results showing edge detection.

A larger array was not possible due to the heavy memory and computational requirements of the simulations. In any case, as the binary edge image only depends on the immediate neighborhood of each pixel, a 256 $\times$ 256-px image was divided into 32 $\times$ 32-px subimages. Each subimage was mapped into the array, on which we incorporate an additional row and column at the bottom and rightmost sides, respectively. This allows for taking into account the neighbors of the pixels at the edges of every subimage. The outcome can be seen in Fig. 5. The discharge interval was  $T_d = 70$ ns. Notice that the binary edge image is available just after this discharge has finished, independently of a possible subsequent anisotropic diffusion. According to the simulations described in the previous section, a first estimation of  $V_{\text{off}}$  for such interval would be 0.25V. This leads to a percentage of false/missing edge locations

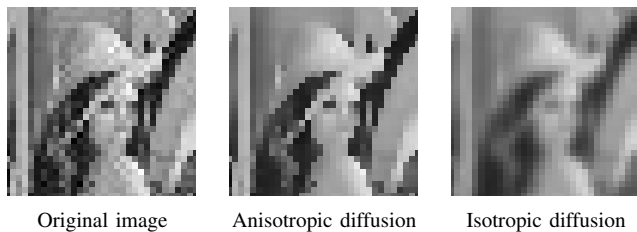


Fig. 6. Image enhancement achieved by anisotropic diffusion.

of 2.52% with respect to an ideal array extracting edges without error. However, it is still possible to find the voltage  $V_{\text{off}}$  better emulated by the network. To this end, we have swept  $V_{\text{off}}$  from 0V up to 0.3V in the ideal array. For each value of  $V_{\text{off}}$ , the output image is compared to the image provided by our non-ideal array. The minimum error percentage, 0.68%, is achieved for  $V_{\text{off}} = 0.15\text{V}$ . That is to say, the edge threshold implemented by our array, whose first approximation was 0.25V, is really 0.15V. Under mismatch conditions, making use of the MOSFET statistical models provided by the manufacturer, the minimum error is 2.78% for  $V_{\text{off}} = 0.14\text{V}$ . Notice that this calibration of  $V_{\text{off}}$  can be easily realized off-chip as a previous step of further analysis of a scene. Once the edge detection functionality is confirmed, let us move on to the other one: image enhancement. The dynamics of the entire RC network is now involved in the final output. This prevents us from making use of images of higher resolution in the same way as for edge detection. The original  $32 \times 32$ -px noisy Lena image mapped into the array is first shown in Fig. 6. The second image corresponds to the output represented by the voltages  $V_{ij}$  after enabling anisotropic diffusion for  $t = 50\text{ns}$ . The value of the elementary capacitor of the RC network simulated is 1pF whereas the dimensions of the elementary MOS resistor are 0.15/1. According to the design methodology described in [6], these elements render a time constant  $\tau = 118\text{ns}$ . Consequently, the width of the Gaussian filter applied is  $\sigma = \sqrt{2t/\tau} = 0.92$ . It can be seen that this filter is adequate to remove the noise while preserving the contrast of the image. For the sake of comparison, we also show the resulting image after applying the same filter via isotropic diffusion. The noise is removed, but at the cost of clearly worsening the contrast. Finally, we must remark that the proposed circuitry can easily accommodate global process variations by adjusting both  $t$  and  $T_d$ . The value of  $t$  will be determined by the equivalent resistance of the elementary MOS resistor at the point of the design space considered [6]. The adjustment of  $T_d$  at that same point will permit to get an estimation of  $V_{\text{off}}$ . We have tuned these parameters for the different corners of the technology in order to carry out the same processing as that of Fig. 6. The maximum RMSE of any of the resulting images with respect to their counterpart for typical conditions is only of 2.7% for the ‘FF’ corner, being  $t = 44\text{ns}$  and  $T_d = 28\text{ns}$  at this corner. Concerning mismatch variations, we have performed 10 Monte-Carlo simulations of the array. Again, we have compared the output images of each simulation with that of Fig. 6, finding a maximum RMSE of 3.7%. These results demonstrate the flexibility and robustness

Author [Ref.]	Tech. ( $\mu\text{m}$ ) / Year	Area ( $\mu\text{m}^2/\text{cell}$ )	Power ( $\mu\text{W}/\text{cell}$ )	Processing time ( $\mu\text{s}$ )
Yu et al [10]	2 / 1992	7500	2.5	< 10
Schemmel et al [12]	0.6 / 2002	2736	$\sim 30$	< 2
Poikonen et al [13]	0.13 / 2010	$\sim 150$	$\sim 50$	$\sim 0.08$
This work	0.13 / 2012	< 3000	0.002	$\sim 0.1$

TABLE I  
COMPARISON WITH PREVIOUSLY REPORTED IMPLEMENTATIONS.

of the design when it comes to address the unavoidable non-idealities of the manufacturing process. Finally, Table I summarizes the main reported features of other implementations of anisotropic filtering. It can be seen that the circuitry described in this paper presents a power consumption four orders of magnitude below the lowest consumption previously reported in the literature. However, it is not very competitive in terms of area usage in its current version. As mentioned previously, a much more area-efficient implementation will be finally incorporated in the targeted smart CMOS imager.

## V. CONCLUSIONS

The implementation of sensing-processing vision chips in 3-D TSV technologies demand low-power building blocks to make the most of the additional computational power available without dramatically rising the power consumption. Focused on this issue, we present an ultra-low-power massively parallel processing array for image enhancement and edge detection. Its operation is based on anisotropic diffusion. In addition to its energy efficiency, the array stands out for its programmability and robustness against process variations.

## REFERENCES

- [1] G. Campardo, G. Ripamonti, and R. Micheloni, Eds., *Proceedings of the IEEE*, vol. 97, no. 1, 2009.
- [2] R. Courtland, “ICs grow up,” *IEEE Spectr.*, vol. 49, no. 1, pp. 33–35, 2012.
- [3] A. Zarandy, Ed., *Focal-plane Sensor-Processor Chips*. Springer, 2011.
- [4] Z. Lin, M. Hoffman, N. Schemm, W. Leon-Salas, and S. Balkir, “A CMOS image sensor for multi-level focal plane image decomposition,” *IEEE Trans. Circuits Syst. I*, vol. 55, no. 9, pp. 2561–2572, 2008.
- [5] A. Lopich and P. Dudek, “A SIMD cellular processor array vision chip with asynchronous processing capabilities,” *IEEE Trans. Circuits Syst. I*, vol. 58, no. 10, pp. 2420–2431, 2011.
- [6] J. Fernandez-Berni and R. Carmona-Galan, “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.
- [7] P. Perona and J. Malik, “Scale-space and edge detection using anisotropic diffusion,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629–639, 1990.
- [8] C. Mead, *Analog VLSI and Neural Systems*. Addison-Wesley, 1989.
- [9] J. Harris, C. Koch, and J. Luo, “A two-dimensional analog VLSI circuit for detecting discontinuities in early vision,” *Science*, vol. 248, no. 4960, pp. 1209–1211, 1990.
- [10] P. Yu, S. Decker, H.-S. Lee, C. Sodini, and J. Wyatt, “CMOS resistive fuses for image smoothing and segmentation,” *IEEE J. Solid-State Circuits*, vol. 27, no. 4, pp. 545–553, 1992.
- [11] J. Athreya and P. Gregson, “1.5V resistive fuse for image smoothing and segmentation,” *Elect. Letters*, vol. 33, no. 10, pp. 851–852, 1997.
- [12] J. Schemmel, M. Karlheinz, and M. Loose, “A scalable switched capacitor realization of the resistive fuse network,” *Analog Integrated Circuits and Signal Processing*, vol. 32, no. 2, pp. 135–148, 2002.
- [13] J. Poikonen, M. Laiho, and A. Paasio, “Anisotropic filtering with a resistive fuse network on the MIPA4k processor array,” in *Int. W. on Cellular Nanoscale Networks and Their Applications*, 2010.