IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 4, 2020, accepted June 17, 2020, date of publication June 22, 2020, date of current version July 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004184

RRAM Crossbar-Based In-Memory Computation of Anisotropic Filters for Image Preprocessing

FAKHREDDINE ZAYER[1], BAKER MOHAMMAD[1], (Senior Member, IEEE), HANI SALEH[1], AND GABRIELE GIANINI[2,3]

[1]System on Chip Center, Electrical Engineering and Computer Science Department, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[2]EBTIC, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates
[3]Computer Science Department, Università degli Studi di Milano, 20133 Milan, Italy

Corresponding author: Baker Mohammad (baker.mohammad@ku.ac.ae)

This work was supported in part by the Abu Dhabi Education and Knowledge (ADEK) Award for Research Excellence under Award AR18-094, and in part by the Khalifa University of Science and Technology under Award RC2-2018-020.

ABSTRACT Anisotropic-diffusion is a commonly used signal preprocessing technique that allows extracting meaningful local characteristics from a signal, such as edges in an image and can be used to support higher-level processing tasks, such as shape detection. This paper presents a fully scalable CMOS-RRAM architecture of an edge-aware-anisotropic filtering algorithm aimed at computer vision applications. The CMOS circuitry controls the scale-space image data to perform pseudo-parallel in-memory computing and nonlinear processing through RRAM crossbar. The arithmetic operations for in-memory computation of brightness gradients are efficiently accumulated to produce the enhanced image in several iterations. The proposed architecture uses single RRAM as a computing and storage element to perform both arithmetic operations and accumulations. Thanks to the in-memory computation, memory accesses and arithmetic operations are reduced by 64% and 92%, respectively, compared to traditional digital implementations. This, in turn, results in a potential reduction of power and area costs of about 75% and 85%, respectively. The processing time is also reduced by 67%.

INDEX TERMS Scale-space image, RRAM crossbar, in memory computing, image enhancement, anisotropic diffusion.

I. INTRODUCTION

In image processing and machine learning tasks the images must often be enhanced in a preprocessing phase – for instance to reduce noise and suppress undesired textures, while, at the same time, preserving and highlighting some other structures. In fact, images typically contain semantically meaningful features, together with irrelevant details: often the former can be qualitatively characterized as local extrema, the pixels along an edge within an image are an example. A difficulty in transforming this qualitative observation into a robust algorithm for feature detection is the following: in order to characterize an extremum some derivatives must be computed over a neighborhood, but there is typically no a priori information about the reference scale at which the differences should be computed. Without such a reference scale, features could be mistaken for noise or vice versa.

The associate editor coordinating the review of this manuscript and approving it for publication was Abdullah Iliyasu.

The scale-space technique introduced by Witkin [1] consists of trying many different scales and looking for the persistence of a derivative change across a range of scales. It prescribes the generation of coarser-resolution images from an original input image by convolving the latter with a Gaussian kernel of variable width (the standard deviation acts as scale parameter). This family of images, ideally obtained across a continuum of scales is called a scale-space image. The scale-space image is then collapsed into a tree description, which is further refined by applying a stability criterion to spot features that persist over large changes in scale. This approach has been used widely and represents a preprocessing step for many higher-level algorithms: edge detection is used as a basis for shape detection, object detection and a variety of early vision tasks [2]. In some tasks, however, once the persistent features have been detected, another issue has to be considered: the original versions of the scale-space technique leave some indeterminacy about the location that should be assigned to the feature (e.g. an edge) in the filtered

output image. To address this problem, Perona and Malik [3] – based on the observation that Gaussian smoothing can be considered as the result of a diffusion process [4], [5] – proposed an anisotropic diffusion approach: the location-dependent diffusion coefficient of the corresponding formulation is set to promote intra-region smoothing in preference to inter-region smoothing. This approach uses a parabolic (i.e. diffusion-like) differential equation as an evolution equation for the system, it considers the original image as the initial state of the process and the steady-state of the process (or the state of the system after a given number of evolution steps) as the output filtered image [6].

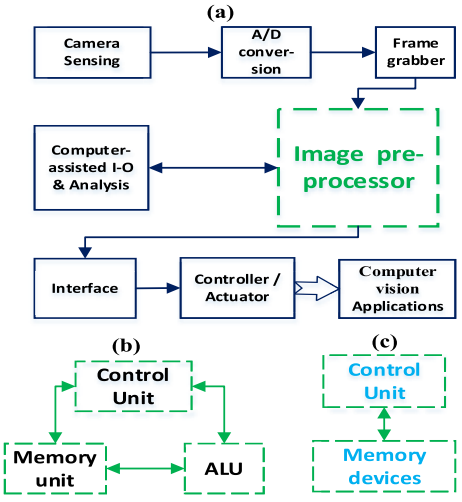

FIGURE 1. a) Machine vision systems. b) Conventional and c) In-memory computing, architectures for image preprocessing.

In practice, anisotropic diffusion (AD) filtering has been widely used for implementations into many computer vision systems for robotics and artificial intelligence (AI) applications as shown in Fig. 1a). Traditional hardware implementations of image processing use (see Fig. 1b) a high-precision number representation system, and pipelines that include a high computational cost algorithm in the preprocessing [7]–[9]. Such solutions require a high-performance microprocessor, high-power, and area costs. Fully dedicated hardware solutions that implement such a preprocessing are limited by poor flexibility and low reusability. They typically target a well-defined application. By contrast, dedicated and programmable approaches can be combined in the design of hybrid architectures: those can better reflect the structure of the image-processing pipeline, which consists of multiple tasks with different computational burdens: they support different degrees of programmability, parallelization, and iterative adaptability. At present, the best-suited solution for the design of flexible, low-complexity and low-power image processing schemas consists of embedding a reduced-instruction-set computer (RISC) engine with dedicated units for intensive computation tasks, such as the in-memory computing (IMC) architecture shown in Fig. 1c).

In the present work, by contrast, we propose an efficient architecture for AD filtering based on a Resistive Random Access Memory (RRAM) crossbar topology, which performs in-memory computation and yields comparable flexibility but higher efficiency. RRAM is a promising candidate for the next-generation non-volatile data storage technology due to its excellent properties including simple structure, symmetric and asymmetric memristive behavior, with pulse/DC sweep voltage supplies [10], nonlinear passive resistance, high storage density, low power consumption, high switching speed and long retention [11]. Owing to its non-volatility, memristive multi-bit based designs, it has found various applications with a scalable and power-efficient analog and digital circuits, [12]–[16]. The most prominent scalable integration of RRAMs is the resistive crossbar memory arrays, in which cross-point are used as storage elements and/or vector-matrix multiplication (VMM) using Ohm's law for multiplication and Kirchhoff's current law for summation [17]. Thanks to the capability of carrying on in-memory computation, coupled with the ability to realize an efficient crossbar topology, the RRAM based circuits perform better than the conventional CMOS memory circuits.

The main contribution of this paper is the innovative use of the memristive crossbar to compute local intensity changes (i.e. compute and store brightness differences between neighboring pixels, i.e. spatial gradients) and to accumulate information through a pseudo-parallel in-memory computing scheme. The processing element, RRAM, also acts as a nonlinear pixel-based resistive state transition that results in averaging and smoothing at the edges. This yields to low power and efficient on-chip solution of the AD implementation for image enhancement. In short, the relevance of the proposed design lies in the capability of the RRAM crossbar in reproducing the AD process thus performing edge-detections and noise-reduction, by a computationally-efficient hardware realization.

We demonstrate the algorithm in image processing aimed at robotics and AI applications. Simulations were performed with 8-bit, 4-bit quantization, and equivalent 4-bit RRAM, respectively. Detail Spice and numerical simulations show comparable results using RRAMs in terms of effectiveness in noise reduction and edge enhancement with efficient dedicated HW performances.

The rest of the paper is organized as follows: Section II defines the scale-space technique and the anisotropic diffusion process for image enhancement. Section III describes the proposed method and the corresponding scalable design architecture and implementation of the AD algorithm. Section IV presents the RISC comparison and a discussion. Conclusions are drawn in Section VI.

II. SCALE-SPACE BASED ANISOTROPIC DIFFUSION

A. THE SCALE-SPACE TECHNIQUE

The standard scale-space technique for vision systems tries to characterize ''semantically meaningful'' features, such as edges, as local changes in the derivatives, that are found to be persistent across the multiple-scale representation [1], [3]. The original scale-space technique [3], when aimed at finding

the edges in a gray-level image would work as follows. The original gray-level image is used to generate other gray-level images, each corresponding to a reference scale. Formally, the algorithm, from an original image $I_0(x, y)$ generates a set of derived images $I(x, y, t)$ obtained by convolution of the original image with a Gaussian kernel $k(x, y; t)$ of variance; larger's correspond to images of coarser resolutions:

$$I(x, y, t) = I_0(x, y) * k(x, y; t) \tag{1}$$

where, $*$, is the convolution operation. The application of the Gaussian convolution has the effect of blurring the image. Consider however two areas of different brightness: if the difference is high enough, then, despite the smoothing, the boundary between regions will determine a noticeable difference at many resolutions. A jump in the image brightness at a given resolution t will yield a peak in the first derivatives and a sign change in the second derivative. Thus, on a 1D signal, finding the zero-crossing points of the second derivative, across different scales in the coarse-to-fine direction will allow to draw a tree-like or nested contour structure, and to identify persistent features, present across scales. Equivalently, in a 2D signal, one can look for the zeros of the Laplacian.

The stability criteria admit several variants, however, even once the persistent features have been detected, a problem arises: one has to determine which is the location that should be assigned to the feature (e.g. an edge) in the filtered output image. The true location of a boundary at a coarse scale in the standard scale-space technique is not directly available in the coarse scale image due to Gaussian blurring [4]. This means that the edge locations in the output image could be shifted from their original locations [1]. To address this problem several approaches were considered and some turned out to be complex and/or computationally costly [18].

B. ANISOTROPIC DIFFUSION

Perona and Malik [3], changed completely the perspective on this problem, based on the observation – due to Hummel *et al.* [4], Hummel [5] – that the Gaussian convolution can be considered as the result of an isotropic diffusion process. They proposed to pass from an isotropic diffusion process – that would equally wash out edges and regions non containing edges – to a controlled anisotropic diffusion process: they prescribed that the location-dependent diffusion coefficient should be set so as to promote intra-region smoothing in preference to inter-region smoothing, thus preserving edges.

Formally, the anisotropic diffusion process is described by the following equation:

$$I_t = Div\left(c\left(x, y, t\right) \nabla I\right) = c\left(x, y, t\right) \Delta I + \nabla \mathrm{c}\left(x, y, t\right) . \nabla I \tag{2}$$

where Div is the divergence operator, $\nabla(\cdot)$ and $\Delta(\cdot)$ are the gradient and Laplacian operators with respect to the space variables, respectively, whereas $c(x, y, t)$ is the diffusion coefficient. The diffusion equation (2) is a special case of a more general class of elliptic equations such that all the maxima of the solution of the equation in space and time belong to the original image, thus fulfilling the so-called causality criterion [5], a consistency requirement.

Suitable intra- and inter-region smoothing can be obtained choosing the diffusion coefficient as a monotonically decreasing function of the gradient of the brightness function, i.e. a decreasing $c(x, y, t) = f(\|\Delta I(x, y, t)\|)$ such that $f(0) = 1$. Indeed, in an approximately flat region, where the gradient ΔI is close to zero, a diffusion coefficient close to 1, grants the maximum diffusion and blurring, whereas in proximity to the edges, where the gradient increases considerably, the diffusion is inhibited. In this way, the diffusion process will essentially take place in the interior of regions, and not affect the region boundaries where the amplitude of the function is large.

Achieving the desired accuracy in edge preservation with the AD process can be computationally expensive if one has to move the information from memory to the arithmetic logic unit (ALU) for every pair of pixels. With respect to this burden, this paper presents an efficient solution based on the hardware friendly, in-memory computing RRAM structure.

III. PROPOSED RRAM CROSSBAR-BASED IN-MEMORY COMPUTING ARCHITECTURE

Along with the recent trends in Internet-of-Thing (IoT) to satisfy specialized hardware systems to be more processing capable than ever, while at the same time satisfying an ultra-low power budget, reconfigurable in-memory processing is the key element for achieving efficient, feasible and practical image processor. However, the Perona and Malik scheme [3] allowed both steering and scaling of an anisotropic bilateral filter. However, in large images, basis and local filters are largely numerous. These filters are non-separable which causes the huge increase of the high power and computational cost in real time applications. Decomposition of the bilateral AD filter into two line filters in non-orthogonal directions [19] was proposed. Choosing an axis to decompose the filter along turns out to be extremely efficient from a computing perspective. In a practical setting, not knowing the axis of orientation for each pixel poses a problem. Therefore, a large number of filters are usually applied at different scales and orientations, and the maximum response per pixel over all the filters is accumulated. Applying a large number of filters commonly requires a significant amount of computing resources. Although several efficient FPGA implementations have been presented in the literature for separable as well as non-separable filters, research on the oriented-filter implementation on an FPGA and/or ASIC is limited [18], [20]. For large windows, several decompositions are used, for example, [21] approximates a large circularly symmetric filter by octagons. An oriented Gaussian smoother [22] was proposed for an efficient FPGA implementation. They decomposed the 2-D filter into 1-D filters and then used pipelining to obtain higher throughput. Only a single orientation has been applied for multiple orientations and a multiple filters in parallel

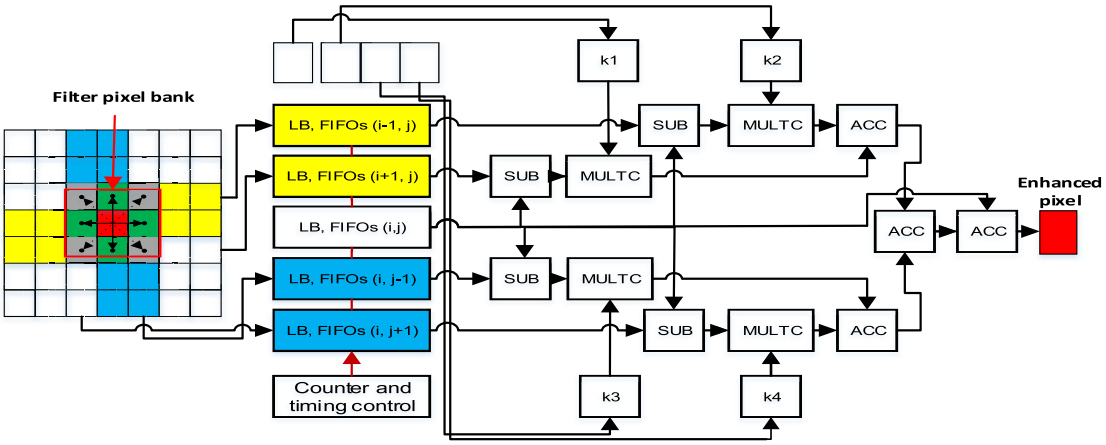

FIGURE 2. Common bilateral AD filter and its needed resources for hardware implementation.

are required. This limits the applicability of these filters for robust image enhancement.

Fig. 2 shows a common anisotropic bilateral filter design and the needed resources for the hardware implementation following a nearest neighbor's discretization of the Laplacian operator as defined in (3);

$$I_{i,j}(n+1) = I_{i,j}(n) + \sigma \cdot \sum_{d=1}^{D} \left[k_{i,j}(d) \cdot \nabla I_{i,j}(d)\right](n) \quad (3)$$

where gradient changes in n iterations are as $\nabla I_{i,j}(d) = \sum I_{i,j}(d) - I_{i,j}$, on $d \in [1, D]$ neighbor's orientations, $k(d)$ are time-constant averaging factors. For the sake of simplicity, the neighbor orientation number is $D = 4$ in Fig. 2. However, in order to perform spatial parallel operations, the image information stream has to be split into multiple sub-streams. An (i, j), $(i \pm 1, j)$, and $(i, j \pm 1)$ memory line buffers (LB) and FiFOs were used to access the required number of the needed lines for processing the targeted pixel window as shown in Fig. 2. The buffer depth depends on the number of pixels in each line of the frame. For an image with N × M pixels, $N \times M \times 4$ subtractions (SUB) for gradient differences, $N \times M \times 4$ multiplications (MULTC) for k(d) time-constant averaging, and $N \times M \times 4$ accumulation (ACC) processes are needed.

In the next sub-sections, signal processing by means of pixel intensity data-quantization for multi-level RRAM functionality is adopted. Work modules for RRAM-based in-memory computation of the local scale-space gradient changes, HW-friendly, spatial-edge-aware filter and nonlinear averaging for smoothing at the edge are described for image reconstruction and enhancement.

A. THE ALGORITHM

A comprehensive signal processing roadmap is adopted for the in-memory image processing. Pseudocode for the proposed method is presented in Algorithm one. However, the continuous analog scale space intensities in $N \times M$ image are sampled into $L_{samp} = 2^{n_{bit}}$ (i.e., 16 samples). A 4-bit scale-space pixel intensity, p_int following x, y directions (i.e, $p\_int(N, M)) \in [0, 255]$ pixel's intensity is quantized and encoded for RRAM in-memory processing, by means, with respect to the time programming voltage range $vmax_{low} : v_{step} : vmax_{high} = [0 : 0.15 : 2.4]\, V$ (i.e., 16 resistive levels). After processing the local pixel through the RRAM crossbar, the analog image is then reconstructed to show the enhanced image at each iteration.

B. RRAM-BASED IN MEMORY COMPUTING ARCHITECTURE

Fig. 3 summarizes the details for the logical architecture to implement the bilateral AD algorithm using RRAM technology. The RRAM model is carefully chosen for the design of an experiment perspective. High speed, multi-level characteristics in the SET process, endurance, and nonlinear switching behavior are supported by the used RRAM technology [14]. Analytically, we slightly modified the model [23] by assuming a low voltage conduction and a symmetric SET operation in both directions to account for the negative and positive neighbor pixel differences for simulating both in-memory computations of the local gradient changes and storage. Explicitly, a temperature dependent Arrhenius dependence of the progressive filament formation, i.e,. $dx/dt = x_0\, exp(-v(t)/kT)$, where $x(t)$ is a state variable, x_0 is a pre-exponential factor, $v(t)$ is the applied voltage for conduction and T is the temperature. The exponential derivative is used to account for the nonlinear SET transitions, which has been widely used for modeling many filament based RRAM devices [24], [25]. However, many RRAM devices with symmetric behavior in both direction and different degree of energy/ latency performances have been demonstrated. For instance, TiOx-based resistive switching device [26] has shown 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme. Programming schemes, strategies and new materials are proposed to implement the symmetric and gradual nonlinear resistive behavior in RRAMs, see for instance, [27], [28]. In fact, multi-level cell operation is simulated by modulating the maximum voltage

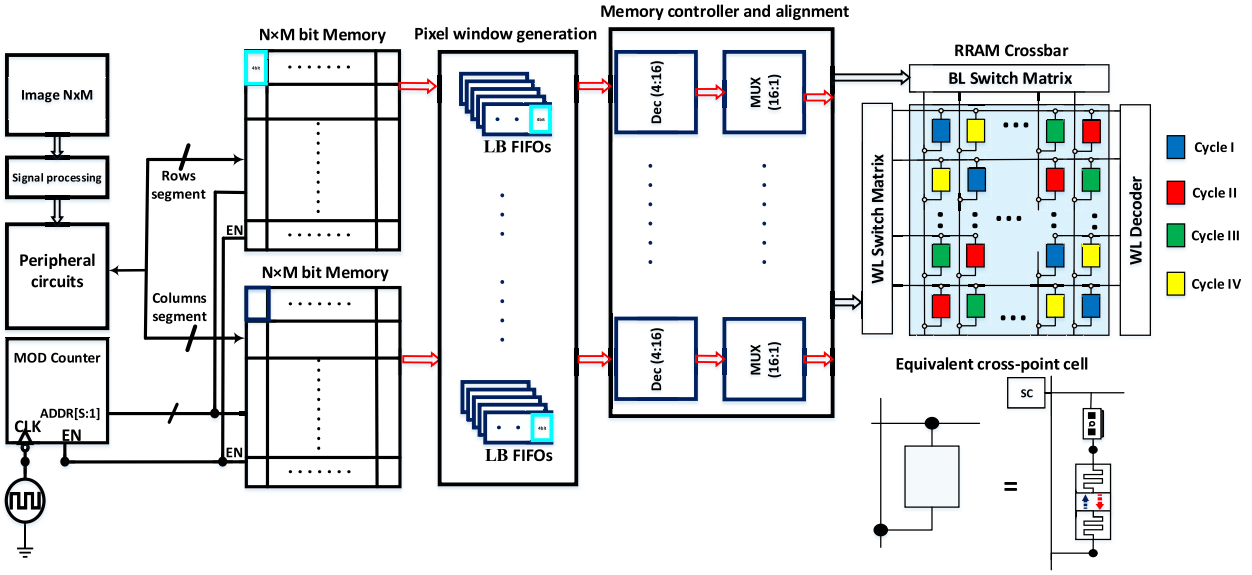

FIGURE 3. The proposed architecture used for in-memory computation of the intensity changes and nonlinear averaging in such direction through N × M cross-point RRAM cells of the crossbar. In the insert, the RRAM cell and its equivalent cross-point with a selector to enable conductance changes in pseudo parallel processing (i.e., four cycles) following the diagonal lines as indicated by the colors on the example of 4 × 4 RRAM crossbar.

of the forward sweep voltage. The produced multi resistive states resulting from the SET operation [14] are used to encode the pixel bit streams.

The pulse generation scheme is performed using a filter pixel window generator and a set of instructions to map pixel values to their corresponding pulse voltage amplitude. However, a two $N \times M$ bit memory for i, j bit planes, containing a serial step pulse-like signal image information are used to drive a set of memory controller and alignment, which control the RRAM crossbar with a 4-bit quantized data bus. Usually, LBs are used to trigger a dedicated filter windo [20]. The length of the buffer line depends on the number of pixels in a considered line, L, from the image. The outputs of the pixel windows are stored in FIFOs (first in-firs-out) structure together with an enable; maximize the pipeline implementation where each pixel is processed per clock cycle. The FIFO interprets the L 4-bit address and outputs the data contained at this address to drive the inputs of a decoder, which control the RRAM crossbar with a 4-bit block size data bus through MUXs. A clocked MOD-N counter, which drives the input address of the memory banks to step through a sequence of instructions, transmitting the specific rows and columns segments to in-memory compute the intensity change with the set of pseudo parallel RRAM as mentioned with colors/cycles on the RRAM crossbar in Fig. 3. To note, the same cycles are used to generate and store pixel filters in the FIFOs. While the difference of neighbor pixels is performed with the application of the corresponding voltage pulses on word, WL, and bit, BL lines, the in-memory processing across cross-point cells is performed only in $(N + M)/2$ cycles where the crossbar dimension is $N \times M \in 2N$. This is equivalent to the write/read, row by row in classical RRAM crossbar designs [29]. At each cycle, only one cross-point cell per line and per column is selected. A very low threshold voltage is investigated, for instance, excellent selector performance was presented in [30]. It is assumed to not have an impact on the intrinsic behavior of the selected RRAM cells. The MOD-N counter is used in this design to help reduce the number of clocks required to address the memory banks so that only a single clock is required instead of a complex system of clocks. The N × M-bit memory units interpret the n-bit address and output the data contained at this address to drive the switch matrixes through the memory controller module.

The synchronous programmed $1 \times N$, WL, and $1 \times M$, BL, line switches are used to program alternatively the devices in order to compute differences of in such neighbor pixel locations in selected diagonal lines. It consists of transmission gates that are connected to all the WLs, with W_1 to W_N control signals of the transmission gates stored in the $N \times M$ scale space image segments. In conductance gradient update, the input signal is loaded to W_1 to W_N / B_1 to B_N, which decides the WLs/BLs to be connected to either the read voltage or ground following the selected RRAM cell in such row and column. In this way, the read voltage that is applied at the input of transmission gates can pass to the WLs/BLs, and the conductance gradients are updated in many iterations using multiple clock cycles in parallel. WL decoder is used for fully parallel signal input to the crossbar rows and columns. The crossbar WL decoder has an additional feature to activate all the programmed WLs per cycle. It is constructed by attaching a controlled power switches, SC, follower circuits to every output row of the decoder. This makes all the internal power switchers transparent for conductance changes following the diagonal lines of the crossbar, which enables pseudo-parallel programming of several diagonal RRAM cells of the crossbar. If all WL decoder lines are open, the crossbar WL decoder

Algorithm 1 Algorithm Pseudocode of RRAM-Based in Memory Computing Using Nonlinear Anisotropic Diffusion for Image Enhancement

procedure Signal processing and pixel intensity data quantization
  for grey image(N, M) **do**
    $p_{int}(x) \leftarrow (p_{int}(N))$
    $p_{int}(y) \leftarrow (p_{int}(M))$
    $p_{int}(x, y) \leftarrow$ analog signal $(p_{int}(x), p_{int}(y))$
      for all $p_{int}(x, y)$ **do**
        $L_{samp} \leftarrow 2^{n_{bit}}$
  $v\left(x, y; 0 : modL_{samp} : 2.4\,(V)\right) \leftarrow p_{int}\left((x, y)\,;\right.$
  $\left.0 : modL_{samp} : 256\right)$
      end for
  end for
end procedure
procedureIn memory-compute and nonlinear processing
  Repeat
    for 1: N iterations **do**
      $V_{WL} \leftarrow v_o\,(\Delta t_i)$, $V_{BL} \leftarrow -v_{neighbors}(\Delta t_i + \Delta t_0)$
      $G_O = f\,(V_{WL} \leftarrow v_o\,(\Delta t_0)$,
      $V_{BL} \leftarrow zero\,(\Delta t_0))$
      G_N:$f\,(V_{WL} \leftarrow v_o\,(\Delta t_2)$,
      $V_{BL} \leftarrow -v_N\,(\Delta t_1 + \Delta t_0)) + G_O$
      G_S:$f\,(V_{WL} \leftarrow v_o\,(\Delta t_3)$,
      $V_{BL} \leftarrow -v_S\,(\Delta t_2 + \Delta t_0)) + \Delta G_N$
      G_E:$f\,(V_{WL} \leftarrow v_o\,(\Delta t_4)$,
      $V_{BL} \leftarrow -v_E\,(\Delta t_3 + \Delta t_0)) + \Delta G_S$
      G_W:$f\,(V_{WL} \leftarrow v_o\,(\Delta t_5)$,
      $V_{BL} \leftarrow -v_W(\Delta t_4 + \Delta t_0)) + \Delta G_E$
    end for
  $\Delta t_i \leftarrow \Delta t_i + \Delta t_0$
Until image enhanced
  $G_{Acc}(N) \leftarrow \sigma \times G_W(N)$
end procedure
Procedure reconstruction of analog image
      for all $G_{Acc}(n, x, y)$ **do**
        $p_{int}\,(n, x, y) \leftarrow G_{Acc}\,(n, x, y)$
      end for
end procedure

will activate all the WLs no matter what input address is given, as the case of cycles I and II. In fact, the design is tolerant with the sneak path problems and electrical and/or thermal coupling that limit the crossbar size. Such a reliable structure is a flexible design by enabling parallel computing.

Level shifters $V_{BL}\,(t + \Delta t)$, see Fig.4b, define the local pixel to be generated, while the directions of diffusion process are specified by the followed pulse stream to be generated and applied on the top/bottom of the RRAM cells to perform in-memory computation of the brightness diffusion. To this end, stored conductance data are accumulated from the four directions (a read operation are enabled after each in-memory computed orientation) with the targeted RRAM cross-point

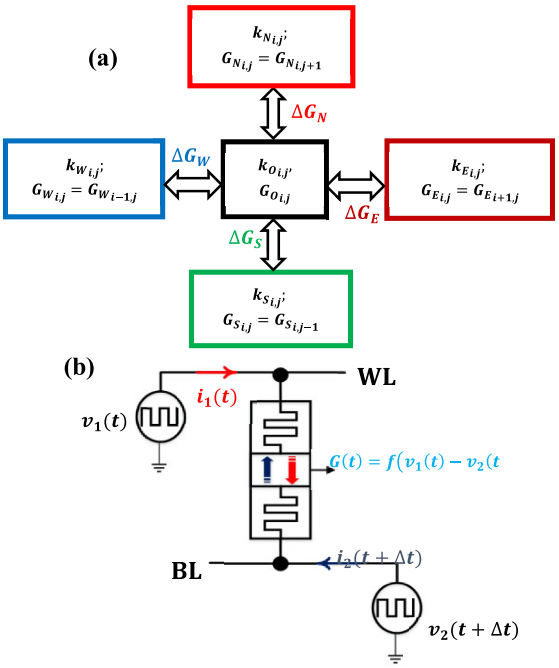

FIGURE 4. The structure of the in memory computational scheme for implementing the physical architecture (see Fig. 3). a) One node of the crossbar and its four neighbors. b) Equivalent RRAM based scheme for the in memory computation, accumulation and averaging.

on the same design that produces the enhanced image in a number of iterations, achieving therefore different compromises between accuracy and information locality. This block diagram is fully scalable; the control block can be adapted to control any number of steps for the computation.

C. IN-MEMORY COMPUTATION OF THE BRIGHTNESS CHANGES

In order to reflect the anisotropic filtering (3) described in the previous subsections, the RRAM cell is considered as an IMC element of brightness changes between neighbor pixels and a nonlinear element to produce smoothing at the edges. We assume that the brightness estimation of the image pixels in each direction is presented by the accumulated and stored output conductance $G\,(x, y) = I(x, y, t)$, at each cross-point of the RRAM crossbar as a function of the programmed input voltage on top and bottom electrodes, V_{TE} and V_{BE} respectively, which are read as a function of the selected WL and BL in $\Delta t = T_2 - T_1$ time, respectively as;

$$\nabla G\,(x, y, t) = \int_{T_1}^{T_2} f(v_{WL} - v_{BL})dt, \tag{4}$$

The associated write pulse conduction $i\,(t) - v(t)$ characteristics produce the nonlinear kernel diffusion coefficients $k\,(x, y, t) = c(x, y, t)$. Equation (2) can be discretized on the RRAM crossbar array shown in Fig. 3. However, edge detection and scale-space formulations are proposed following the tensor scheme presented in Fig. 4a).

The 4-nearest neighbor's discretization scheme is now defined as (5) following the equivalent two-terminal

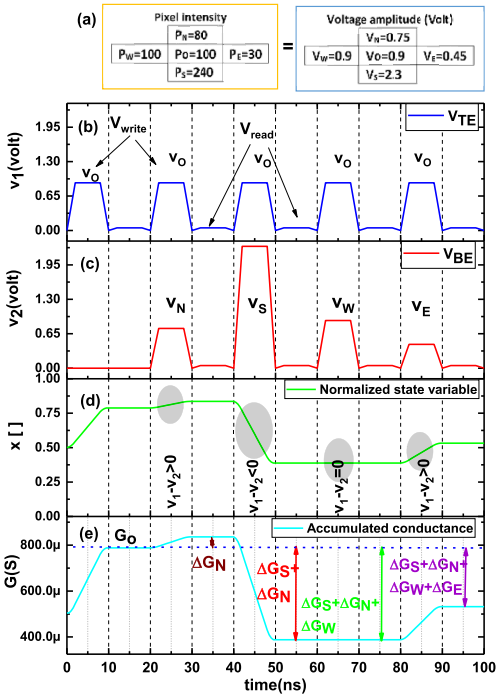

FIGURE 5. In memory-computation of the gradient change between neighbor pixels and their accumulation in one iteration (an example of a cross-point RRAM). a) Origin (Po) and its neighbor pixel intensities (PN,PS, PW, PE) and their corresponding modulated pulse signals (Vo, VN, VS, VW, VE), respectively. b) and c) The applied programming voltage to WL and BL cross-point RRAM. d) The normalized internal state variable and its change, historic accumulation according to the differences in the applied voltage at its terminals. e) The resulted readout conductance, $G(t)$.

memristive circuit shown in Fig. 4b);

$$G_{i,j}(n_k) = G_{i,j}(n_{k+1}) + \sigma \begin{pmatrix} f\left(\nabla_N G_{i,j}(n_k)\right) + \\ f\left(\nabla_S G_{i,j}(n_k)\right) + \\ f\left(\nabla_W G_{i,j}(n_k)\right) + \\ f\left(\nabla_E G_{i,j}(n_k)\right) \end{pmatrix} \quad (5)$$

where $0 < \sigma < 1/4G_r$ is a stability factor for the proposed scheme, $G_r \approx 1mS$ is the dynamic conductance range. $n_1, n_2 \ldots n_K$ stands for the iteration number. N, S, E, W are the mnemonic indices for North, South, East, West. ∇ indicates nearest-neighbor differences (6). The conductance gradient at each location is modulated as an absolute conductance values of its projection along with RRAM cells in the crossbar row and column directions as follows;

$$\begin{aligned} \nabla_N G_{i,j}(t) &= \left\| G_{i-1,j}(t) - G_{i,j}(t) \right\| \\ \nabla_S G_{i,j}(t) &= \left\| G_{i+1,j}(t) - G_{i,j}(t) \right\| \\ \nabla_E G_{i,j}(t) &= \left\| G_{i,j+1}(t) - G_{i,j}(t) \right\| \\ \nabla_W G_{i,j}(t) &= \left\| G_{i,j-1}(t) - G_{i,j}(t) \right\| \end{aligned} \quad (6)$$

Fig. 5 shows an example of a given origin pixel (Po) and its four neighbors (PN, PS, PW, PE) and the in-memory processing (i.e compute the gradient conductance and accumulation) as a function of the applied voltage supplies (V_O, V_N, V_S, V_W, V_E), associated to the mentioned pixel's tenser scheme. Cadence Virtuoso with the RRAM spice model is used to simulate the multi-level cell operations by modulating the maximum voltage of successive write pulses forwardly, each one is followed by a non-destructive read pulse, as shown in Fig. 5b), c). The write pulses (1ns rise/full times and 8ns pulse width) are ranging from 0V to 2.4V by a step of 0.15V. The write/read scheme is used for reliability assessment as RRAMs suffer from device variability in write processes. A smooth nonlinearity associated with the conductance state transitions is produced as shown in Figs. 5d) and 5e). The memristive state variable and the resulted readout conductance are accumulated and nonlinearly evolved as a function of the applied pulse-associated neighbor pixel intensities over time. Following the amount of neighbor pixel intensity, the in-memory computed brightness changes define the locality of the edges in such direction. For $V_{BE} > |V_{TE}|$, the output conductance is slightly increased while for $V_{BE} < |V_{TE}|$ is decreased and remains equal when the neighbor pixels have the same amount of brightness. However, conductive gradients are updated at every iteration following the history of the accumulated brightness gradient $G_{i,j}(t)$. The gradient can be computed on different neighbor cells in the crossbar structure achieving different compromises between accuracy and locality. $f(\cdot)$ is also updated at every iteration as a function of the level of brightness gradient $G_{i,j}(t)$ given at specific neighbor pixel based-pulse amplitudes.

$$\begin{aligned} G_{Oi,j}(t) &= f\left(\left\| G_{O,i,j}(t) \right\|\right) \\ G_{Ni,j}(t) &= f\left(\left\| \nabla_N G_{i,j}(t) \right\|\right) + G_{Oi,j}(t) \\ G_{Si,j}(t) &= f\left(\left\| \nabla_S G_{i,j}(t) \right\|\right) + G_{Ni,j}(t) \\ G_{Wi,j}(t) &= f\left(\left\| \nabla_W G_{i,j}(t) \right\|\right) + G_{Si,j}(t) \\ G_{Ei,j}(t) &= f\left(\left\| \nabla_E G_{i,j}(t) \right\|\right) + G_{Wi,j}(t) \end{aligned} \quad (7)$$

This discretized formulation maintains the property of the continuous derivative (2), which means the total amount of brightness in the image, is preserved. Additionally, the flux of brightness through each column and row directions of the crossbar (see Fig. 3) only depends on the brightness values at the two nodes defining it, which makes the proposed design a natural choice for analog VLSI implementations.

D. CASE STUDY: ALGORITHM IMPLEMENTATION

The algorithm is validated on a grey scale image. The original image, I_0, is corrupted by random noise with standard deviation, $r_n = 30$ to form the noisy image, $I_n = double(I_0) + 30 \times randn(size(I_0))$, note that $randn(\cdot)$ generates random numbers, and hence the results is different for every instance. To get around this problem, generate I_n once and then use it for all experiments to obtain consistency when comparing the methods for RRAM based AD algorithm validation. Fig. 6 presents a noisy and filtered pepper image using the RRAM based AD algorithm. Fig. 7 shows the pixel intensities vs their image locations with respect to the original image. PSNR (peak signal to noise ratio) and SSIM (structural similarity) metrics are used to quantify the effectiveness of using RRAM nanotechnology for a hardware friendly and

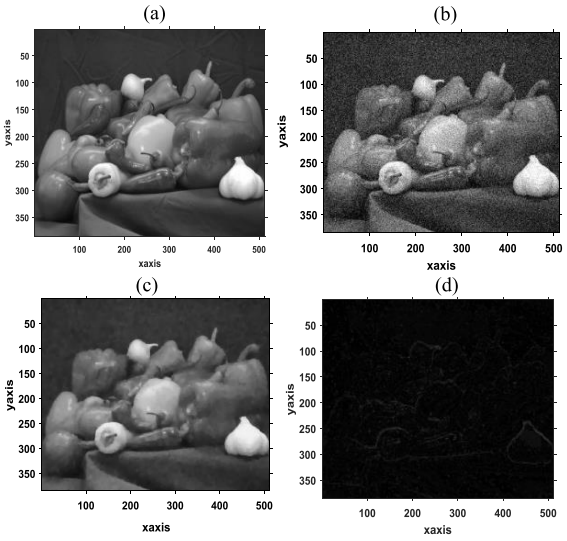

FIGURE 6. a) Original, b) Noisy input and c) restored pepper images and d) Edge preservation using RRAM based AD algorithm.

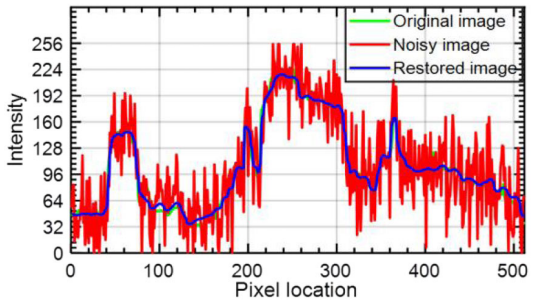

FIGURE 7. Pixel intensity vs locations for the original, noisy and filtered image, respectively.

computationally efficient AD implementation. The results in Fig. 8 and the comparative table of the algorithm performances using 8-bit, 4-bit quantization and 4bit RRAM presented below proves that the 4-bit RRAM performs the same performance as digital 4-bit quantization with a slight increase of the number of iterations. This is not a drawback while the computation of intensity difference between neighbor local pixels and their accumulations are performed internally in a single cross-point RRAM cell. In our application, for each filter window (origin pixel and its neighbors), the origin pixel will be written one time to the cell, then multiple iterations are needed to in-memory compute the differences between the origin pixel and its neighbors for accumulate operation. Besides, while the image is naturally analog, write energy for computing *difference* operations is relatively small as the intensity difference between neighbor pixels is low. Moreover, the number of resistive levels and the nonlinear behavior in state transitions define the adaptive number of iterations for reducing image noise and perform the image filtering and edge restoration. Accordingly, device variation may increase also the iteration number resulting in a slight increase in latency and energy but not the image quality.

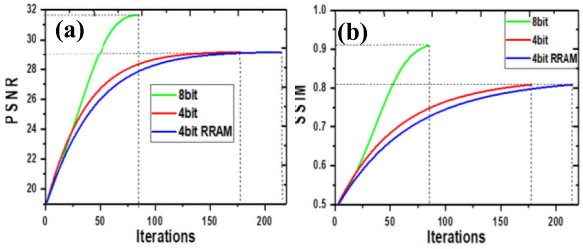

FIGURE 8. a) PSNR and b) SSIM Performance metrics used to evaluate image quality using 8-bit, 4-bit and 4-bit RRAM, respectively.

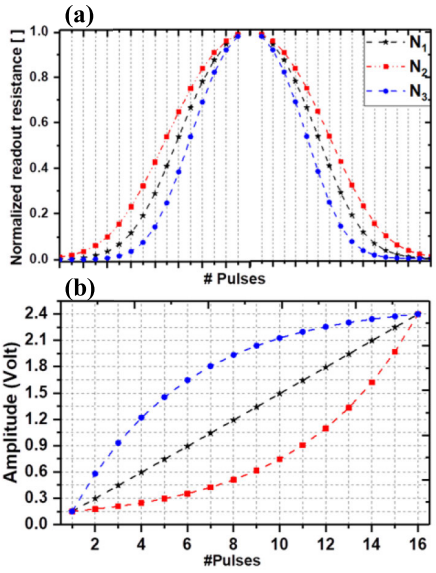

FIGURE 9. a) Different memristive nonlinearity shapes using, b) different voltage programming scheme.

E. IMPACT OF NONLINEARITY ON PRESERVING EDGES

The nonlinear behavior of the memristive characteristics is tested using three voltage programming schemes as shown in the inset of Fig. 9. The write pulse amplitude linearly or nonlinearly increases with the pulse number. The conductance (or resistance) dependent voltage amplitude (G-V or R-V) with a nonlinearity, N_i, can be specified as the conductance ratio of the modeled RRAM device. This results in a different shape of nonlinearity (i.e, N_1, N_2, N_3) along with the transition between resistive states like-pixel intensities (see Fig. 9). PSNR, SSIM, and correlation measures are performed to figure out the edge preservation using the RRAM based AD algorithm. As shown in Table 2, any RRAM-based nonlinearity shape could produce comparable results of image enhancement and edge preservation with respect to 8bit and state of the art of AD algorithm [20]. Yet, a smooth nonlinearity shows slightly better results.

IV. PERFORMANCE ASSESSEMENT AND DISCUSSIONS

In this section, a comparative analysis of the proposed RRAM architecture is presented, along with conventional FPGA based solutions. However, RRAM shows a small footprint, CMOS compatible BEOL process, fast accessing compared to FLASH, etc. [31]. The RRAM cell size is, $C_z = 4F^2$.

TABLE 1. Comparison between Algorithm performances using 8-Bit, 4-Bit and 4-Bit RRAM, respectively.

#bit / Metrics	8-bit	4-bit		4-bit RRAM	
#iterations	85	85	178	85	215
PSNR	31.6	28.3	29.1	27.9	29.1
SSIM	0.91	0.75	0.81	0.73	0.81

TABLE 2. Impact of memristive nonlinearity on the image enhancement using the 4-Bit RRAM quantization.

Nonlinearity / Metrics	N_1	N_2	N_3
PSNR(dB)	29.1	29.8	28.6
SSIM	0.81	0.87	0.79
Correlation	0.97	0.98	0.96

TABLE 3. Parameters used to calculate required resources and performance assecement for local pixel computations.

Architecture / Parameter	Conventional design-based FPGA solutions	RRAM-based solution
Mem_{acc}	9 [33]	5
$Ar.op$	12 [33]	1 (in-memory compute)
Pro_{time} (ns)	1200 (8-bit) [20], [32] 800 (4-bit quantization)	100
A (F^2)	120 (SRAM blocks) [31] 320 (other blocks) [35].	4+260 (WL, BL switch matrices +WL decoders)

While the cell sizes of SRAM, Dynamic Random Access Memory (DRAM) and STT-RAM are $A = 120F^2$, $A = 6F^2$, and $A = 20F^2$, respectively [31]. An image size, $I_z = 256 \times 256$ is chosen for performance evaluation (i.e, memory access(Mem_{acc}), number of arithmetic operations (Ar.op), processing time (Pro_{time}), power consumption(P) and area cost(A)). Table 3 shows the parameters for evaluating the required resources for a single local pixel computations using traditional FPGA solutions and the proposed RRAM implementation.

The total number of memory access to compute the required arithmetic operations for a given local pixel dependent neighborhood information in traditional FPGA solution is given as follows;

$$Mem_{acc} = Mem_{acc}\,(v_{in}) + Mem_{acc}(k_{i,j}) \tag{8}$$

where $v_{in} = v_{in}(\text{P}_\text{o}) + 4v_{neighbor}$, is five access times for the encoded neighbor pixel-input signals. $Mem_{acc}(k_{i,j})$ is the 4-time register accesses as the simplest best case of a constant-time nonlinear averaging process as shown in Fig. 2. While for the RRAM integration, 5 memory accesses are required to process in-memory the targeted pixel location as shown in Figs. 4 and 8. Based on Fig. 2, the number of arithmetic operations needed for computing using traditional FPGA implementations [20], [32], [33] is calculated as;

$$\text{Ar.op} = \text{SUB}(\cdot) + \text{MULTC}(\cdot) + \text{ACC}(\cdot) \tag{9}$$

TABLE 4. Performance assessments and reduced resources using the proposed RRAM based design w. r. t conventional FPGA implementations.

Architecture / Metrics	Previous HD implementations	RRAM-based AD	RISC comparison/saving
Memory access	589824	327680	64%
Arithmetic operations	786432	65536	92%
Processing time(ns)	1200(8-bit) 800(4-bit)	400	67% 50%
Power (watt)	3.408 (OAD [20]) 7.943 (TF [34]); $I_z = 150 \times 150$	16; $I_z = 400 \times 550$	~75% for $I_z = 400 \times 550$
A(F^2)	3.457.280	67,584	85%

These operations are done naturally using RRAM based in-memory computations (i.e. only the access pulse train is required across its nodes). The RRAM-based IMC, instead of traditional FPGA implementation [32], shows a vast decrease in both; number of memory accesses and arithmetic operations.

The processing time in real-time applications is defined as the required time to process given parallel pixels per clock cycle. For the clock frequency, $F_c = 100MHz$ (period $T_c = 10ns$) used for the FPGA platform on which the experiment processes a local pixel per clock pulse. The image processing time, Pro_{time}, is estimated by assuming best case as fully parallel computing in conventional FPGA structure;

$$Pro_{time} = \#bit \times \Delta t \times n_{cyles} + delays \tag{10}$$

where $\#bit$ is the number of bits in, n, number of cycles to process the accessed input signals for a local pixel processing and Δt is the bit-time information. Delays must be considered for accessing the memory following the pixel-neighbor's locality. As for the RRAM architecture, the pseudo parallelism with four programming cycles described in the previous section leads to 4 pulse timing (100ns per cycle, as shown in Fig. 5).

The calculated power, using RRAM to process image pixels, is defined as;

$$\begin{aligned} P_{RRAM} = {} & G_{i,j} \times v_{i,j}^2 + \text{n} \times \Big(\big[\nabla_N G_{i,j} + \nabla_S G_{i,j} + \nabla_E G_{i,j} \\ & + \nabla_W G_{i,j} \big] \times \big[\nabla_N v_{i,j} + \nabla_S v_{i,j} + \nabla_E v_{i,j} \\ & + \nabla_W v_{i,j} \big]^2 \Big) \end{aligned} \tag{11}$$

where the first term corresponds to write original image on the crossbar RRAM, while the second term corresponds to write the accumulated differences with respect to the applied voltage difference between neighbor pixels in n iteration. Power values are presented in Table 4. However, from the OAD (optimized anisotropic diffusion) [20] and the TF (trilateral filter) [34], a random noise $r_n = 12$ and a number of iterations, $\text{n} = 4$, are used to perform the algorithm computation in hardware for a natural Einstein grayscale image of size 150 × 150. In our case, using pepper image with a size of 400×550, $r_n = 30$ and $n = 185$ parameters are

TABLE 5. Performance metrics/operation comparing various embedded memory candidates for the implementation of ad-based image processing algorithm.

Operation	SRAM [36]			FeFET [36]			RRAM [14], [36]		
	#Cells/Op	Write Energy/Op	Write Latency/Op	#Cells/Op	Write Energy/Op	Write Latency/Op	#Cells/Op	Write Energy/Op	Write Latency/Op
IMC Differences	SP and MUs	~fJ	~1 ns	SP and MUs	--	--	1R	~1 pJ-0.1 pJ	~10ns-100ps
IMC Accumulations	6T[4]			1T	~0.1 pJ	~10 ns			
Nonlinear processing	SP and MUs	--	--	SP and MUs	--	--			

used in our case study. The consumed power for image based-filtering and edge preservation depends greatly on the image size, applied noise, and the iteration number. This explains the greater calculated power when using RRAM, as shown in Table 4. However, using a similar image and parameters, P_{RRAM} is estimated to be 4x less than that consumed in conventional hardware implementation [20], [34]. Thanks to in-memory processing, simply in response to the difference in voltage, P_{RRAM} is expected to be very low compared to conventional computing if hard noise and large image size apply.

As for the area costs in conventional FPGA solutions, additional and optimized areas for the subtraction and weighted averaging could be estimated from neuromorphic processors, for instance, [35]. The RRAM compared to SRAM based design results in a saving of 85% of the area cost. However, peripheral CMOS circuits for RRAM control are critical. For advanced technology node, the ratio area/ delay is still similar because most of the CMOS devices are the smallest pass gates. In fact, the main delay confinement still applies to CMOS technology. With advanced CMOS technologies, area overhead of the memory buffers, which is a trade-off between area and delay time, could be alleviated in RRAM based design because the delay time is smaller since the path is shorter.

As for comparison, SRAM and ferroelectric field-effect transistor (FeFET) are used as a baseline for comparisons (see table V) as these technology-based designs can perform the same operations as RRAM architecture for AD algorithm implementation. However, SRAM is the most reported stable design in term of low energy latency and endurance characteristics in the IMC scope [36]. As for FeFET, a challenging 32 levels of conductance states have been recently demonstrated [37], [38]. But, SRAM, for instance, is a volatile and binary type memory (single bit per cell). FeFET does not allow in-memory compute the difference operation in response to the difference in voltage as it is a transistor based. Therefore, separate processing (SP) and memory units (MUs) are needed to either perform the IMC differences, accumulations operations, or nonlinear averaging when using the above-mentioned technologies. This will cause a larger area and hence exhibit higher delay and energy costs. Moreover, voltage-latency with multi-level behavior of FeFETs [37], [38], are limited by parasitic.

RRAM, despite variability issues that could be alleviated while scarifying more cycle iterations, has comparable metrics and is a suitable solution for the implementation of AD algorithm as the proposed RRAM architecture has more efficient design due to the single cell ability to have multiple level and nonlinear change in response to the difference in voltage.

The scale-spaces generated by the above-described scheme can be a suitable choice to implement the nonlinear diffusion satisfying the set of criteria listed in section II in order to generate the multiscale ''semantically meaningful'' representations of images. The advantages of the proposed design includes; energy-efficient estimated detection accuracy, small on-chip area, and scalability of the memristive crossbar array comparing to traditional CMOS circuits as presented in Table 4. As for the crossbar approach, the total power required for the conductance update according to the write operations is low. Additionally, the proposed memristive system could be scaled with lower leakage current when compared to the conventional CMOS design.

Overall, the existing mixed-signal, FPGA and analog implementations of the edge-aware image enhancement tasks have large on-chip area and high power dissipation drawbacks [39], [40]. Thanks to small on-chip area and scalability of using memristive circuits, the proposed RRAM architecture based image-edge detection and enhancement module is an appropriate solution for image pre-processing. In fact, the scalable design shown in Fig. 3 that enables the in-memory and a self-assisted nonlinear processing results with an ultra-low power, potentially reduced on-chip area and efficiently accelerates the processing time for a particular application and integration into the existing pixel sensors and used for the edge-computing-based AI and robotic applications. Nonetheless, device exploration to meet the algorithm requirements will be investigated in the future. Further analysis including RRAM non-ideality, for instance, the number of resistive states, the dynamic range, variability, energy, and latency will accomplished to show the real impact and the viability of using RRAM for filtering and edge detection based image pre-processing tasks.

V. CONCLUSION

In this paper, a fully scalable and hardware friendly architecture using the nonlinear anisotropic diffusion algorithm for

storing and in-memory processing a given image is proposed. Multi-level characteristics, endurance and nonlinear behavior are supported by RRAM technology. Image pixels are quantized to perform filtering and local edge enhancement with multi-level IMC operations through the RRAM crossbar. Pseudo parallel computing is proposed to accelerate the AD algorithm. The Super self-assisted non-linear processing by means the kernel coefficients associated with the write pulses produce a smoothing at the edges. Brightness gradients following several directions through an adaptive tensor scheme and pseudo parallel processing are conducted in order to satisfy a set of criteria for obtaining ''semantically meaningful'' multiple scale descriptions. Accumulation of these gradients could be read efficiently through the design structure to produce an enhanced image. Results show some huge improvements in terms of power, area overheads and accelerations. The use of RRAM is an efficient solution where the computation cost is a major concern since it reduces the design complexity and speeds up the computation.

REFERENCES

[1] A. P. Witkin, ''Scale-space filtering,'' in *Readings in Computer Vision*, M. A. Fischler and O. Firschein, Eds. San Francisco, CA, USA: Morgan Kaufmann, 1987, pp. 329–332.

[2] T. Lindeberg, *Scale-Space Theory in Computer Vision*. New York, NY, USA: Springer, 1994.

[3] P. Perona and J. Malik, ''Scale-space and edge detection using anisotropic diffusion,'' *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990, doi: 10.1109/34.56205.

[4] R. A. Hummel, B. Kimia, and S. W. Zucker, ''Deblurring Gaussian blur,'' *Comput. Vis. Graph. Image Process.*, vol. 38, no. 1, pp. 66–80, Apr. 1987, doi: 10.1016/S0734-189X(87)80153-6.

[5] R. Hummel, *The Scale-Space Formulation of Pyramid Data Structures*. London, U.K.: Forgotten Books, 2018.

[6] J. Weickert, *Anisotropic Diffusion in Image Processing*. Stuttgart, Germany: Teubner, 1998.

[7] Z. Liang, W. Liu, and R. Yao, ''Contrast enhancement by nonlinear diffusion filtering,'' *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 673–686, Feb. 2016, doi: 10.1109/TIP.2015.2507405.

[8] A. Gabiger-Rose, M. Kube, R. Weigel, and R. Rose, ''An FPGA-based fully synchronized design of a bilateral filter for real-time image denoising,'' *IEEE Trans. Ind. Electron.*, vol. 61, no. 8, pp. 4093–4104, Aug. 2014, doi: 10.1109/TIE.2013.2284133.

[9] S. D. Dabhade, G. N. Rathna, and K. N. Chaudhury, ''A reconfigurable and scalable FPGA architecture for bilateral filtering,'' *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1459–1469, Feb. 2018, doi: 10.1109/TIE.2017.2726960.

[10] W. Xiao, W. Song, Y. P. Feng, D. Gao, Y. Zhu, and J. Ding, ''Electrode-controlled confinement of conductive filaments in a nanocolumn embedded symmetric–asymmetric RRAM structure,'' *J. Mater. Chem. C*, vol. 8, no. 5, pp. 1577–1582, Feb. 2020, doi: 10.1039/C9TC06552K.

[11] F. Zayer, W. Dghais, and H. Belgacem, ''Modeling framework and comparison of memristive devices and associated STDP learning windows for neuromorphic applications,'' *J. Phys. D, Appl. Phys.*, vol. 52, no. 39, Sep. 2019, Art. no. 393002, doi: 10.1088/1361-6463/ab24a7.

[12] F. Zayer, W. Dghais, M. Benabdeladhim, and B. Hamdi, ''Low power, ultrafast synaptic plasticity in 1R-ferroelectric tunnel memristive structure for spiking neural networks,'' *AEU Int. J. Electron. Commun.*, vol. 100, pp. 56–65, Feb. 2019, doi: 10.1016/j.aeue.2019.01.003.

[13] C.-X. Xue *et al.*, ''24.1 a 1Mb multibit ReRAM computing-in-memory macro with 14.6ns parallel MAC computing time for CNN based AI edge processors,'' in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2019, pp. 388–390, doi: 10.1109/ISSCC.2019.8662395.

[14] U. Böttger, M. von Witzleben, V. Havel, K. Fleck, V. Rana, R. Waser, and S. Menzel, ''Picosecond multilevel resistive switching in tantalum oxide thin films,'' 2019, *arXiv:2002.00700*. [Online]. Available: http://arxiv.org/abs/2002.00700

[15] S. Yu, B. Gao, Z. Fang, H. Yu, J. Kang, and H.-S.-P. Wong, ''A neuromorphic visual system using RRAM synaptic devices with sub-pJ energy and tolerance to variability: Experimental characterization and large-scale modeling,'' in *IEDM Tech. Dig.*, Dec. 2012, pp. 10.4.1–10.4.4, doi: 10.1109/IEDM.2012.6479018.

[16] M. Zhao, H. Wu, B. Gao, Q. Zhang, W. Wu, S. Wang, Y. Xi, D. Wu, N. Deng, S. Yu, H.-Y. Chen, and H. Qian, ''Investigation of statistical retention of filamentary analog RRAM for neuromophic computing,'' in *IEDM Tech. Dig.*, Dec. 2017, pp. 39.4.1–39.4.4, doi: 10.1109/IEDM.2017.8268522.

[17] C. Li, C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves, Z. Li, J. P. Strachan, P. Lin, Z. Wang, M. Barnell, Q. Wu, R. S. Williams, J. J. Yang, and Q. Xia, ''Analogue signal and image processing with large memristor crossbars,'' *Nature Electron.*, vol. 1, no. 1, p. 52, Jan. 2018, doi: 10.1038/s41928-017-0002-z.

[18] M. Schaffner, F. Scheidegger, L. Cavigelli, H. Kaeslin, L. Benini, and A. Smolic, ''Towards edge-aware spatio-temporal filtering in real-time,'' *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 265–280, Jan. 2018, doi: 10.1109/TIP.2017.2757259.

[19] J. Geusebroek, A. W. M. Smeulders, and J. van de Weijer, ''Fast anisotropic gauss filtering,'' *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 938–943, Aug. 2003, doi: 10.1109/TIP.2003.812429.

[20] C. Pal, A. Kotal, A. Samanta, A. Chakrabarti, and R. Ghosh, ''An efficient FPGA implementation of optimized anisotropic diffusion filtering of images,'' *Int. J. Reconfigurable Comput.*, vol. 2016, pp. 1–17, Mar. 2016, doi: 10.1155/2016/3020473.

[21] S. Kawada, ''An approach for applying large filters on large images using FPGA,'' in *Proc. Int. Conf. Field-Program. Technol.*, 2007, pp. 201–208, doi: 10.1109/FPT.2007.4439250.

[22] A. Joginipelly, A. Varela, D. Charalampidis, R. Schott, and Z. Fitzsimmons, ''Efficient FPGA implementation of steerable Gaussian smoothers,'' in *Proc. 44th Southeastern Symp. Syst. Theory (SSST)*, Mar. 2012, pp. 78–82, doi: 10.1109/SSST.2012.6195131.

[23] A. Siemon, D. Wouters, S. Hamdioui, and S. Menzel, ''Memristive device modeling and circuit design exploration for computation-in-memory,'' in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5, doi: 10.1109/ISCAS.2019.8702600.

[24] S. Kim, S.-J. Kim, K. M. Kim, S. R. Lee, M. Chang, E. Cho, Y.-B. Kim, C. J. Kim, U.-In Chung, and I.-K. Yoo, ''Physical electro-thermal model of resistive switching in bi-layered resistance-change memory,'' *Sci. Rep.*, vol. 3, no. 1, p. 1680, Apr. 2013, doi: 10.1038/srep01680.

[25] L. Larcher, F. M. Puglisi, P. Pavan, A. Padovani, L. Vandelli, and G. Bersuker, ''A compact model of program window in HfOx RRAM devices for conductive filament characteristics analysis,'' *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2668–2673, Aug. 2014, doi: 10.1109/TED.2014.2329020.

[26] J. Park, M. Kwak, K. Moon, J. Woo, D. Lee, and H. Hwang, ''TiO_x-based RRAM synapse with 64-levels of conductance and symmetric conductance change by adopting a hybrid pulse scheme for neuromorphic computing,'' *IEEE Electron Device Lett.*, vol. 37, no. 12, pp. 1559–1562, Dec. 2016, doi: 10.1109/LED.2016.2622716.

[27] S. Park, A. Sheri, J. Kim, J. Noh, J. Jang, M. Jeon, B. Lee, B. R. Lee, B. H. Lee, and H. Hwang, ''Neuromorphic speech systems using advanced ReRAM-based synapse,'' in *IEDM Tech. Dig.*, Dec. 2013, pp. 25.6.1–25.6.4, doi: 10.1109/IEDM.2013.6724692.

[28] P.-Y. Chen, X. Peng, and S. Yu, ''NeuroSim: A circuit-level macro model for benchmarking neuro-inspired architectures in online learning,'' *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 12, pp. 3067–3080, Dec. 2018, doi: 10.1109/TCAD.2018.2789723.

[29] Y. Jiang, J. Kang, and X. Wang, ''RRAM-based parallel computing architecture using k-nearest neighbor classification for pattern recognition,'' *Sci. Rep.*, vol. 7, no. 1, Mar. 2017, Art. no. 45233, doi: 10.1038/srep45233.

[30] C. Wang, B. Song, and Z. Zeng, ''Excellent selector performance in engineered Ag/ZrO_2:Ag/Pt structure for high-density bipolar RRAM applications,'' *AIP Adv.*, vol. 7, no. 12, Dec. 2017, Art. no. 125209, doi: 10.1063/1.5009717.

[31] Y.-C. Chen, W. Wang, H. Li, and W. Zhang, ''Non-volatile 3D stacking RRAM-based FPGA,'' in *Proc. 22nd Int. Conf. Field Program. Log. Appl. (FPL)*, Aug. 2012, pp. 367–372, doi: 10.1109/FPL.2012.6339206.

[32] T. M. Khan, D. G. Bailey, M. A. U. Khan, and Y. Kong, ''Efficient hardware implementation for fingerprint image enhancement using anisotropic Gaussian filter,'' *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2116–2126, May 2017, doi: 10.1109/TIP.2017.2671781.

[33] K. N. Chaudhury, D. Sage, and M. Unser, "Fast $O(1)$ bilateral filtering using trigonometric range kernels," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3376–3382, Dec. 2011, doi: 10.1109/TIP.2011.2159234.
[34] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," in *Proc. ACM SIGGRAPH Courses (SIGGRAPH)*, Los Angeles, CA, USA, Jul. 2005, p. 5-es, doi: 10.1145/1198555.1198565.
[35] C. S. Thakur, R. Wang, T. J. Hamilton, R. Etienne-Cummings, J. Tapson, and A. van Schaik, "An analogue neuromorphic co-processor that utilizes device mismatch for learning applications," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 65, no. 4, pp. 1174–1184, Apr. 2018, doi: 10.1109/TCSI.2017.2756878.
[36] S. Salahuddin, K. Ni, and S. Datta, "The era of hyper-scaling in electronics," *Nature Electron.*, vol. 1, no. 8, pp. 442–450, Aug. 2018, doi: 10.1038/s41928-018-0117-x.
[37] K. Chatterjee, S. Kim, G. Karbasian, D. Kwon, A. J. Tan, A. K. Yadav, C. R. Serrao, C. Hu, and S. Salahuddin, "Challenges to partial switching of Hf0.8Zr0.2O2 gated ferroelectric FET for multilevel/analog or low-voltage memory operation," *IEEE Electron Device Lett.*, vol. 40, no. 9, pp. 1423–1426, Sep. 2019, doi: 10.1109/LED.2019.2931430.
[38] S. Oh, T. Kim, M. Kwak, J. Song, J. Woo, S. Jeon, I. K. Yoo, and H. Hwang, "HfZrOx-based ferroelectric synapse device with 32 levels of conductance states for neuromorphic applications," *IEEE Electron Device Lett.*, vol. 38, no. 6, pp. 732–735, Jun. 2017, doi: 10.1109/LED.2017.2698083.
[39] P. R. Possa, S. A. Mahmoudi, N. Harb, C. Valderrama, and P. Manneback, "A multi-resolution FPGA-based architecture for real-time edge and corner detection," *IEEE Trans. Comput.*, vol. 63, no. 10, pp. 2376–2388, Oct. 2014, doi: 10.1109/TC.2013.130.
[40] D. Bronzi, F. Villa, S. Tisa, A. Tosi, F. Zappa, D. Durini, S. Weyers, and W. Brockherde, "100 000 frames/s 64×32 single-photon detector array for 2-D imaging and 3-D ranging," *IEEE J. Sel. Topics Quantum Electron.*, vol. 20, no. 6, pp. 354–363, Nov. 2014, doi: 10.1109/JSTQE.2014.2341562.

FAKHREDDINE ZAYER received the B.S. degree in physics and the M.S. degree in materials, nanostructures, devices and microelectronic systems from the Faculty of Sciences, University of Monastir, Monastir, Tunisia, in 2011 and 2013, respectively, and the Ph.D. degree in electronics and computer engineering from the National Engineering School of Monastir (ENIM), University of Monastir, in 2019. He was a Senior Staff Engineer with the Integration of Materials to the System (IMS), Thales Group, France, from 2014 to 2016. He has been a Research Associate in electronic engineering with the Khalifa University of Science and Technology, United Arab Emirates, since 2019. His research interests include compact modeling of nanoscale devices, such as reliability and testability, and computing paradigms, such as neuro-computing and in-memory computing for logic and storage.

BAKER MOHAMMAD (Senior Member, IEEE) received the B.S. degree in ECE from The University of New Mexico, Albuquerque, the M.S. degree in ECE from Arizona State University, Tempe, and the Ph.D. degree in ECE from The University of Texas at Austin, in 2008. He worked for a period of ten years with Intel Corporation, in a wide range of micro-processors design from high-performance, server chips more than 100Watt (IA-64), and mobile embedded processor low-power sub one watt (xscale). He has over 16 years industrial experience in microprocessor design with emphasis on memory, low-power circuit, and physical design. He was a Senior Staff Engineer/Manager with Qualcomm, Austin, TX, USA, for a period of six years, where he was engaged in designing high-performance and low-power DSP processor used for communication and multi-media application. He is currently the Director with the System on Chip Center and an Associate Professor of EECS with the Khalifa University of Science and Technology. He is also engaged in microwatt range computing platform for wearable electronics, WSN focusing on energy harvesting, power management, and power conversion, including efficient dc/dc and ac/dc converters. He has authored/coauthored over 100 refereed journals and conference proceedings and three books. He holds over 18 U.S. patents. He participates in many technical committees with the IEEE conferences and reviews for journals, including TVLSI and the IEEE Circuits and Systems. He has multiple invited seminars/panelist and the presenter of three conference tutorials, including one tutorial on Energy harvesting and Power management for WSN, in 2015 (ISCAS). His research interests include VLSI, power efficient computing, high-yield embedded memory, and emerging technology, such as memristor, STTRAM, in-memory-computing, and hardware accelerators for cyber physical systems. He received several awards, including the KUSTAR Staff Excellence Award in intellectual property creation, the 2009 Best paper Award from Qualcomm Qtech Conference, the IEEE TVLSI Best Paper Award, the 2016 IEEE MWSCAS Myrill B. Reed Best Paper Award, the Qualcomm Qstar Award for Excellence on Performance, the Leadership, the SRC Techon Best Session Papers, in 2016 and 2017, and the Intel Involve from the Community Award for Volunteer and Impact on the Community. He serves as an Associate Editor for the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (TVLSI) and *Microelectronics Journal* (Elsevier).

HANI SALEH received the Bachelor of Science degree in electrical engineering from The University of Jordan, the Master of Science degree in electrical engineering from The University of Texas at San Antonio, and the Ph.D. degree in computer engineering from The University of Texas at Austin. He worked for many leading semiconductor design companies, including a Senior Chip Designer (Technical Lead) with Apple Inc., Intel (Atom mobile microprocessor design), AMD (Bobcat mobile microprocessor design), Qualcomm (QDSP DSP core design for mobile SOC's), Synopsys (designed the I2C DW IP included in Synopys Design Ware library), Fujitsu (SPARC compatible high performance microprocessor design) and Motorola, Australia. He has a total of 19 years of industrial experience with ASIC chip design, microprocessor design, DSP core design, graphics core design, and embedded system design. He has been an Associate Professor of electronic engineering with the Khalifa University of Science and Technology, since 2012. He is currently a Co-Founder and an Active Researcher with the Khalifa University Research Center (KSRC) and the System on Chip Research Center (SOCC), where he led multiple the IoT projects for the development of wearable blood glucose monitoring SOC, mobile surveillance SOC, and AI accelerators for edge devices. He holds over 20 issued U.S. patents and eight pending patent applications. He has over 120 articles published in peer-reviewed conferences and journals in the areas of digital system design, computer architecture, DSP, and computer arithmetic. His research interests include the IoT design, deep learning, AI hardware design, DSP algorithms design, DSP hardware design, computer architecture, computer arithmetic, SOC design, ASIC chip design, FPGA design, and automatic computer recognition.

GABRIELE GIANINI received the M.Sc. degree, in 1992, and the Ph.D. degree, in 1996. He held visiting positions at a number of international institutions, including the INSA de Lyon, France, the University of Passau, Germany, CERN, Geneva, Switzerland, the Fermilab, Chicago, IL, USA, CBPF, Brazil. From 2005 to 2012, he was an Adjoint Professor with the Free University of Bolzano, Italy. Since 2017, he was a Senior Research Fellow with EBTIC, Khalifa University. He is currently an Associate Professor with the Department of Computer Science, Università degli Studi di Milano, Italy. He has coauthored over 200 papers published in internationally refereed journals and conferences. His research interests include machine learning and data analytics applications. Since 2018, he has been an Associate Editor for *Journal Data Science and Engineering (DSEJ)*(Springer), and *Journal of Imaging Science and Technology (JIST)*, Society for Imaging Science and Technology.