Received January 15, 2018, accepted March 23, 2018, date of publication April 2, 2018, date of current version April 23, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2819986

Bit-Plane Extracted Moving-Object Detection Using Memristive Crossbar-CAM Arrays for Edge Computing Image Devices

NAZGUL DASTANOVA, SULTAN DUISENBAY, OLGA KRESTINSKAYA, (Graduate Student Member, IEEE), AND ALEX PAPPACHEN JAMES, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, School of Engineering, Nazarbayev University, Astana 010000, Kazakhstan

Corresponding author: Alex Pappachen James (apj@ieee.org)

ABSTRACT In this paper, we present the hardware implementation of a novel algorithm for moving-object detection, which can be integrated with CMOS image sensors. Bit planes of consecutive frames are stored in memristive crossbar arrays and compared using threshold-logic XOR gates. The resulting outputs are combined using weighted summation circuits and thresholded using comparators, to obtain binary images. A resistive content-addressable memory (CAM) array is used in the output stage to observe the numbers of different object pixels in the first and second pairs of the processed frames, in a row-by-row manner. The CAM array output conveys information on the motion direction and allows for optimal memory utilization through the selective row-wise storage of different bits. The proposed method outperforms the conventional moving-object detection algorithms, in terms of accuracy, specificity, and positive prediction metrics, and performs comparably in terms of other metrics.

INDEX TERMS Moving object detection, bit-plane extraction, memristor, crossbar array, threshold logic gate, edge devices.

I. INTRODUCTION

The detection of object movement is a low-level vision activity that enables the human visual system to learn and extract object features. The moving-object detection techniques use the process of detecting object-pixel changes relative to the surrounding environment [1]. From among the multiple ways of implementing motion-based object detection, the camera-based system is prevalent because of the robustness shown by its higher-level vision processing algorithms [2]. The camera-based object-motion detection systems have been used extensively in a range of industrial applications such as traffic monitoring, people tracking [1], and video surveillance [3]. Motion detection techniques are divided into three expansive categories: background subtraction [4], optical flow [5], and frame differencing [6]. The algorithm presented in this paper relies on the extension of the frame-differencing approach via bit-plane extraction.

An image pixel in digital space can be represented as a bit sequence with each bit plane illustrating the corresponding bit position [7]. The higher order bits contain the most valuable visual data, while the remaining bit planes give less discernible information, such as small background changes [8]. The extraction of bit planes has applications in biometrics (face [9], iris [10], and palm print [11] recognition), image compression [12], and biomedical image retrieval [13]. Since bits have only two values, they can be physically stored in a two-state memory device. One such device is a memristor that acts as a two-terminal resistive switch [14]. Owing to its non-volatility and ability to memorize the resistance states, the memristor has found various applications in analog and digital circuits, as well as in mimicking neuromorphic processes [15]. In digital circuits, a memristor can be utilized as a power-efficient and scalable two-state device.

The most prominent scalable digital applications of memristors are crossbar memory arrays and resistive content-addressable memories (CAMs), in which the memristors are used as storage elements by creating a passive interconnected network. The crossbar topology performs better than the conventional CMOS memory circuits, and is considered to be a successor to the conventional memory and logic configurations [16]. Memristive crossbar arrays have also been used in conjunction with pixel sensors, for the learning and recognition of static images [17]. Another application of memristors is in weighted input circuits, also known as threshold logic gates (TLGs). Inspired by the firing process of neurons, TLGs have recently been implemented using memristors as synaptic weights [18].

In this paper, memristor crossbar networks are used to detect moving objects in a dynamic environment by comparing consecutive frames. This approach is motivated by the need for introducing video analytics into the Internet-of-Things systems. The ability to detect moving objects in vision sensors can help to develop higher order tasks such as pattern recognition and intelligent decision making. The proposed circuit can be used in near sensor processing tasks and be integrated with pixel sensors, can potentially be applied in the fields of security, law enforcement, and military applications. The use of memristive elements in the design ensures the implementation of switching behavior as well as small on-chip area and low power dissipation [19], [20].

The two main contributions of this paper are: (1) a novel algorithm for moving-object detection based on bit-plane extraction, and (2) the innovative use of memristive crossbars and CAM arrays for translating the algorithm to an on-chip solution. The novelty of the algorithm lies in the bitwise manipulation of pixels, resulting in a more computationally efficient method for hardware realization, than the conventional methods. The proposed system can be integrated into the existing pixel sensors and used for the edge computing applications because it does not require additional software to perform the frames comparison.

The rest of this paper is organized as follows: Section II discusses provides the relative background on moving object detection algorithms, memristive circuits and recently proposed moving detection approaches and their hardware implementation. The software and hardware implementation of the proposed method is describe in Section III. The experimental analyses and results are presented in Section IV. Section V provides the explicit discussion on the advantages and drawbacks of the proposed moving object detection implementation and highlights the possible future works. The conclusions are drawn in Section VI.

II. RELATED BACKGROUND

In this section, we provide background on the most relevant moving-object detection algorithms and the memristive circuits used in this paper. The moving-object detection, being a low-level vision activity, requires information processing at pixel levels, and is a natural fit to hardware implementation. The pixel-level analysis requires parallel processing, as the speed is limited by the traditional software implementations.

A. BACKGROUND ON MOVING-OBJECT DETECTION ALGORITHMS

The three major groups of techniques used in the moving-object detection algorithms are: (1) background subtraction, (2) frame differencing, and (3) optical flow. In addition, as we introduce a pixel-based change detection, the existing pixel-based object detection methods are covered in this section.

1) MAJOR MOVING-OBJECT DETECTION TECHNIQUES

a: BACKGROUND SUBTRACTION

Among the existing moving-object detection algorithms, the ones based on background subtraction are popular because of their simplicity and feasibility. The main idea lies in subtracting the object from its background in the current frame, by applying preprocessed background models. The resulting image is further binarized by using a threshold to indicate the moving objects. This method gives excellent results for videos from static cameras; however, dynamic backgrounds and outdoor environments can affect the quality severely. The disadvantages of such methods are mitigated by applying soft clustering and intensity histogram [21] techniques, which however do not solve the issues of shadows and ghost effects. A range of techniques with increased complexities exists, which can improve the background subtraction. These include the Mixture of Gaussians (MOG) background subtraction, subspace learning background subtraction, statistical background subtraction, fuzzy background subtraction, and Robust Principal Component Analysis (RPCA) background subtraction [22], [23] techniques. However, with increasing complexity, the circuit-level implementations become increasingly complex and unviable for low-power implementations.

b: FRAME DIFFERENCING

Frame differencing implies subtracting the current video frame from the previous frames. Dependent on the number of frames considered, two- and three-frame differencing methods are deployed. In two-frame differencing, the current and previous video frames are subtracted, and the resultant image is thresholded. In three-frame differencing, the current, previous, and next frames are considered [6]. Three-frame differencing is an extension of the conventional two-frame differencing approach, and it improves the overall performance and detection speed [6]. Frame-differencing methods can be improved by additional computational processes such as automatic thresholding or use of linear operators [24]; however, issues related to occlusion, illumination, and background noise arise.

c: OPTICAL FLOW

One of the computationally complex approaches for moving-object detection is the optical flow method. In this approach, two successive frames are processed, and each pixel is considered as a vector. The displacement of each pixel over time, compared to the previous pixel, is calculated [5]. Owing to the complexity of calculations, three dimensional images or matrices (as RGB images) are transformed to two dimensional matrices (as grayscale images), and further using the brightness constancy assumption, the pixel intensity:

$$f(x, y, t) = f(x + dx, y + dy, t + dt), \tag{1}$$

By employing the Taylor series for the right-hand side of Eq. (1), and making some changes to the obtained equation, we obtain an interpretation for the optical flow equation as:

$$f_x u + f_y v + f_y = 0, \tag{2}$$

It can be also represented in a vector form as:

$$\nabla f \vec{v} = -f_t, \tag{3}$$

where ∇f is the brightness intensity spatial gradient, $\vec{v}$ is the speed vector of a pixel, and f_t is the time derivative of brightness intensity. Equation (2) is commonly used in optical-flow estimation, and is known as the gradient constraint [5]. Optical-flow equations are solved using the Lucas–Kanade or Horn–Schunck methods [5].

2) PIXEL-BASED CHANGE DETECTION METHODS

There is several software-based methods and algorithmic implementations of moving object detection. According to the analyzed input features and approach used to sort input visual data, the change detection methods can be categorized into pixel based, transformation based, texture based, structure based methods and fusion of several methods [25]. The pixel-based change detection methods include image differencing,rationing and regression, change vector analysis, median filtering-based background formation and pixelwise fuzzy XOR operations [25], [26]. As a part of the proposed method, we use the XOR operation, which already has been proven to be efficient for frame differencing and pixel-based change detection approaches, especially in fuzzy XOR-based moving object detection methods [25], [27]. In addition, the XOR-based algorithms are the least complex in terms of hardware implementation, comparing to the other pixel-based change detection methods.

B. BACKGROUND ON RELATED MEMRISTIVE CIRCUITS

In the hardware implementation proposed in this paper, three major circuit concepts are used, which are: (1) crossbar array, (2) threshold logic gates, and (3) content-addressable memories.

1) CROSSBAR ARRAYS

The crossbar-structure design for memories has been used extensively since the 1970s. The basic structure of a crossbar array incorporates a mesh network of wires with switches at the junctions [28]. Bits are stored in these switches in the form of ON/OFF states, and the state is written or read by applying an appropriate voltage. Crossbar systems are inferior to conventional CMOS memory designs, in terms of power and energy efficiency. These disadvantages primarily arise from the inherent switching characteristics. With the incorporation of memristors as switches, the mentioned issues have widely been addressed, drawing considerable attention [29]. The ability of memristive devices to exhibit the switching behavior and non-volatile properties make the memristors appropriate solution to solve the scalability problems and implement the crossbar for large scale applications [19], [20]. The drawback of using memristive crossbar arrays lies in the generation of sneak paths when current flows through undesired memristors during the read phase [30]. These sneak paths lead to decreased accuracy and higher power consumption; therefore, the sneak path issue is of primary concern in the simulations [31].

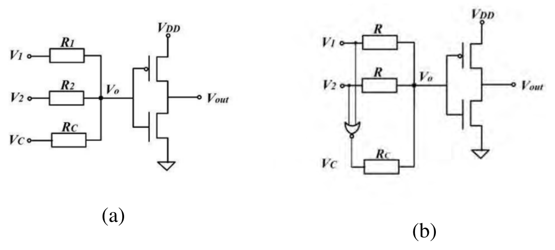

FIGURE 1. Memristive linear threshold gate in (a) *NAND, NOR* and (b) *XOR* configuration.

2) THRESHOLD LOGIC

Threshold logic has found many fields of application, including digital gates, computing, and bio-inspired processing [32]. Threshold logic gates (TLGs) have been considered as being successors to the conventional logic gates [33]. The ability of threshold logic design to accomplish complex logic functions by using the intrinsic properties of TLGs allows for fewer gates and levels, compared to standard logic designs [34]. Threshold logic design has attracted attention owing to its potential to implement multiple additions, multiplications, divisions, and sorting. Furthermore, TLGs have found applications in modeling neural networks and brain-inspired processes, including learning, adaptive, and pattern-recognition systems [34]. In general, the transfer function of a linear TLG is given by:

$$f(x_1, \ldots, x_n) = \begin{cases} 1, & \sum_{i=1}^{n} w_i x_i \geq T \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where x_i is the input variable, w_i is the weight corresponding to input i, and T is the threshold given by an integer number.

Several attempts have been made to build TLGs with memristors [32]. One of the promising TLG structures is presented in [18]. In this configuration, memristors are used as input weights, and CMOS acts as a threshold logic device. By utilizing the resistive switching mechanism in memristors, the TLG can be set in the NAND, NOR, or XOR configurations [18], [32]. The implementation of NAND, NOR, or XOR configurations of TLG proposed in [32] are shown in Fig. 1.

3) CONTENT-ADDRESSABLE MEMORY

Content-addressable memory (CAM) is a type of memory in which the input data are compared to the data stored in the memory. The CAM has a significant advantage in terms of speed, as the comparison is achieved in a single clock cycle. Although they demonstrate fast performances, the CAM circuits have very high power consumptions, which limit their applications. An additional disadvantage of using CAM is the increased silicon area [35]. The inputs to the CAM array, which are compared with the stored patterns, are provided by search lines. The outputs of CAM are provided by match lines, and are dependent on the similarities between the input and stored patterns. Memristors have been proposed as efficient alternatives for transistors in CAMs, to address the issue of high power consumption and small area scalability. A promising design for a memristive CAM cell was introduced in [36], which utilized two transistors and two memristors ($2T-2M$ configuration) for a single cell.

C. RELATED WORKS

The recently proposed moving object detection algorithms are based on various approaches, such as Bayesian approach [37], automatic K-means clustering algorithm [38], background modeling approaches with fusion of color and texture features [39], Zernike moments [40], low-rank and invariant sparse decomposition [41], modifications of simultaneous localization and mapping methods [42], deep sequence learning [43], improved and modified low-rank modeling methods [44], principal component analysis [45] and various modifications of background subtraction methods [46]. One of the main problem of such algorithms is the complexity. The complexity of the existing moving object detection algorithms makes their hardware implementation using analog circuits difficult or impossible. Even if several analog and mixed-signal implementations exist, high power dissipation, large area and lack of scalability of such systems do not allow the CMOS analog circuits based method compete with software-based and completely digital solutions. In turn, the analog implementation can make the video data processing more efficient and faster.

There several digital implementations of moving object detection based on various algorithms on Digital Signal Processor (DSP) [47] and Field Programmable Gate Array (FPGA) [48]–[51]. The mixed-signal implementation of moving object detection is proposed in [52]. The system contains analog CMOS circuits for edge detection inspired by biological vision system, digital circuit for generalization of motion signals and software-based signal analysis using microcomputer. The other mixed signal implementation of the moving object detection on a static background is shown in [53]. The module relies mostly on digital processing and uses the capacitor as the analog storage for previous video frame which enables the possibility of frame difference operation.

There are few analog integrated circuit approaches for detecting the static objects against moving background. One of such approaches represented in [54] is based on lower animal vision and focuses on edge detection. The circuit contains latch, velocity detection and comparator circuits implemented using $1.2\mu m$ CMOS process. However, the detection of the moving objects using analog hardware approach has not been implemented yet. There other moving object detection system relying mostly on analog circuits is proposed in [55]. However, the circuit is based on CMOS technology which makes it large and inefficient in terms of power. In addition, the scalability problem of moving object detection components exists in the existing CMOS-based implementations. The proposed memristive-CMOS based circuit can be one on the solutions to the scalability problems. Recently, the memristor based fast moving object detection system has been introduced in [56]. The memristive object detection system uses pixel-based change detection approach and shows high accuracy results. However, this system requires an additional sensing circuit to be integrated with the traditional CMOS sensors.

III. PROPOSED METHOD

The pseudocode of the proposed method is presented in Algorithm 1. Two consecutive images are converted to grayscale, sliced into bit-planes, and bit-wise compared. The higher order bits used for the comparison are further combined and processed to obtain the result. The process is repeated for the next two consecutive frames. The algorithm is verified using target videos from the VISOR database [57].

Algorithm 1 Algorithm Pseudocode for Object Detection

```
1:  repeat
2:      procedure BitExtraction($frame_i, frame_{i+1}$)
3:          for both frames do
4:              $gray\_frame \leftarrow convert\_to\_gray(frame)$
5:              for all bit planes do
6:                  $b\_p \leftarrow get\_bit\_planes(gray\_frame)$
7:              end for
8:          end for
9:      end procedure
10:     procedure XOR($frame_i, frame_{i+1}$)
11:         for both frames do
12:             $result\_b\_p \leftarrow XOR(b\_p_i, b\_p_{i+1})$
13:         end for
14:         for high order bit planes do
15:             $gray\_result \leftarrow combine(result\_b\_p)$
16:         end for
17:     end procedure
18:     $i \leftarrow i + 1$
19: until frames_end
```

The input color images, corresponding to $frame_i$ and $frame_{i+1}$, are acquired and are converted to 8-bit grayscale images ($gray\_frame$). Eight constituting bit planes are extracted from the grayscale pixel values ranging from 0 to 255. As a result, eight binary images are formed (b_p), which are later used in the comparison process. Bit-plane extraction is achieved by:

$$\lfloor \frac{Y}{2^k} \rfloor mod2 = a_k, \tag{5}$$

where Y is the grayscale pixel value, k is the bit number, mod is a modulo operation, and a_k is the bit value of the

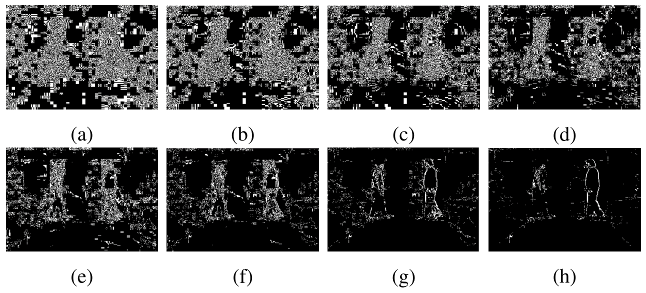

FIGURE 2. Comparison of bit planes via XOR operation: (a) LSB, (b) 1st bit, (c) 2nd bit, (d) 3rd bit, (e) 4th bit, (f) 5th bit, (g) 6th bit, and (h) MSB. The (a), (b), (c) and (d) refer to the small insignificant background changes, while (e), (f), (g) and (h) refer to the significant changes.

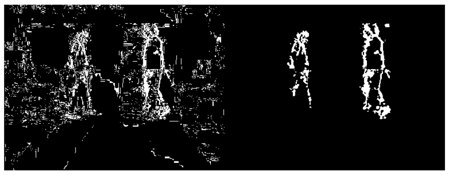

FIGURE 3. Detection of the moving object. Original image is acquired from the combination of the XORed bit planes and the thresholded image.

corresponding bit number. $\lfloor\cdot\rfloor$ is the floor operation represented by:

$$\lfloor x \rfloor = m \iff m \leq x < m + 1, \tag{6}$$

Since the lower order bit planes do not convey important information, only the higher order bit planes are stored for further processing. This will also contribute to effective memory utilization.

The corresponding stored bit planes of consecutive frames are compared via an XOR operation (*result_b_p*), which is illustrated in Fig. 2. The XOR operation is used as a distance metric, to calculate the pixel differences between bit planes. The XOR operation is given by:

$$a_k \oplus b_k = c_k, \tag{7}$$

where a_k, b_k, and c_k are the k-th bit values of the previous frame pixel, current frame pixel, and resultant bit value, respectively.

After comparing the bit planes, the higher bit planes are merged together once again to obtain a grayscale image by:

$$\sum_{k=4}^{7} 2^k * c_k = Y, \tag{8}$$

where k is the bit number, c_k is the bit value, and Y is the resultant grayscale pixel value. The obtained grayscale image is further thresholded to detect the object. The example of the grayscale and corresponding binary image frames is shown in Fig. 3.

The translation of the proposed algorithm to a circuit is illustrated in the block diagram shown in Fig. 4. Before the processing and comparison of the video frames, the external

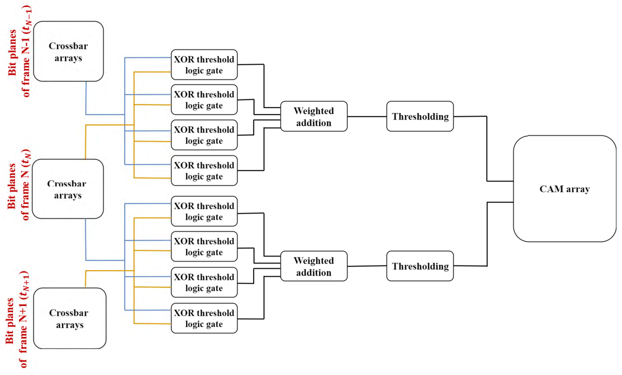

FIGURE 4. Hardware block diagram of the proposed circuit. Bit planes are stored and read from crossbar arrays, compared via memristive threshold logic gates, processed, and compared in CAM arrays.

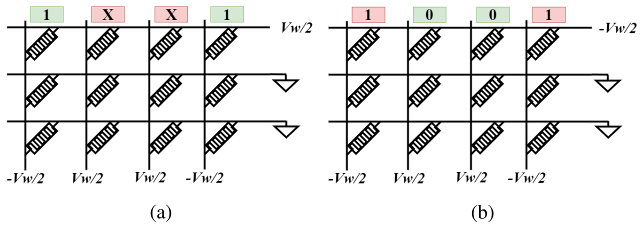

FIGURE 5. Memristor crossbar array write cycle. (a) "1" valued bits are written in the first cycle, (b) "0" valued bits are written in the second cycle.

control and pre-processing unit selects the video frame samples, which are stored in a crossbars. Memory is of crucial importance in the proposed circuit, as the integration with pixel sensors requires high scalability and reduced power consumption for standalone operation. Integration of conventional SRAM memories faces the problems of high leakage and static power dissipation. As a result, the extracted bit planes are stored in crossbar arrays, which use memristors as memory elements. The stored elements in the crossbar arrays representing the higher order bit planes if the consecutive frames in time are compared using XOR TLGs. Therefore, only the planes representing 4th bit, 5th bit, 6th bit and MSB are stored and fetched into the XOR gates. Then, the weighted summation of 4 XOR outputs of the consecutive frames (for example, frame $N-1$ at time t_{N-1} and frame N at time t_N) is performed. Next, the obtained values are thresholded and saved into the CAM array. In the next cycle, the same operation is performed for the frames N and $N+1$, however the CAM array is set to the comparison mode. In this mode, the thresholded output of the frames N and $N+1$ is compared to the previously stored output of the frames $N-1$ and N.

Bit planes in memristive crossbar arrays are stored using two row-by-row successive write cycles, which are illustrated in Fig. 5. The control unit is used to control the applied voltage and switch between elements during the write cycles. The memristor states are set by applying positive or negative write voltages, which are greater than the memristor's

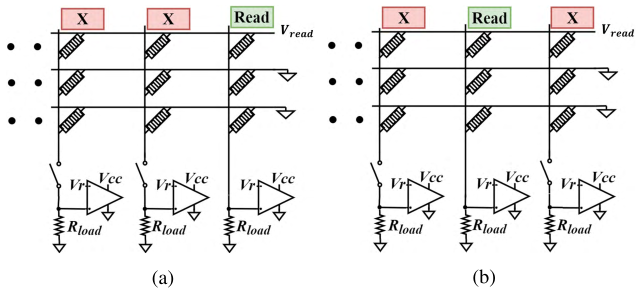

FIGURE 6. Memristor crossbar array read cycle. (a) Reading the first element in the first row, (b) reading the second element in the first row.

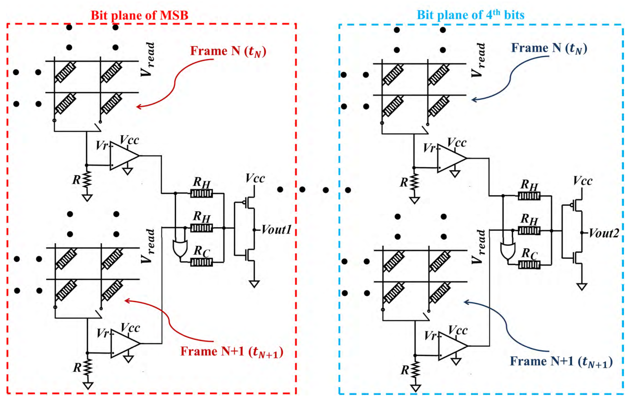

FIGURE 7. Integration of memristive crossbar arrays with memristive threshold logic gates (TLGs) in XOR configuration.

threshold voltage ($V_w > V_{thresh}$). During the first cycle, bits of value "1" are written into the memristors in the form of low resistances, while "0" valued bits remain idle. This is achieved by applying a positive write voltage across the memristor terminals. During the second cycle, bits of value "0" are written in the form of high resistances, while the already stored "1" bits remain idle.

The stored bits are read by applying a read voltage, which is smaller than the threshold voltage, to avoid resetting the switches ($V_{read} < V_{thresh}$). The bits are extracted via comparator circuits embedded in the crossbar's columns. The comparator circuit comprises a load resistor and op-amp along with the reference voltage. Depending on the memristor's resistance, either "0" or "1" is obtained at the output of the comparator circuit. To deal with the sneak path problem, memristors are read element-wise after applying the read voltage to each row. This is accomplished by introducing an isolating switch between the crossbar array and the comparator stage. The crossbar read cycle is depicted in Fig. 6.

In the next stage, memristive TLGs are used for realizing the XOR operation. The bits corresponding to the bit planes of two consecutive frames stored in the crossbar arrays are taken as inputs to the TLGs. The integration of the memristive crossbar arrays and the TLGs is shown in Fig. 7. For all memristive elements in the array, the process of XOR comparison is done sequentially switching from element to element. The switching between the memristors is done by external control circuit. The logic gates are setup to implement the XOR configuration illustrated in [32], which requires the control voltage V_c to correspond to the NOR output of the two input voltages. The output from the TLG corresponds to the compared bit value to form compared bit planes.

Depending on the application and the limitations of the system where moving object detection module is integrated, different memristor crossbar configurations should be considered to achieve the trade-off between the processing speed, power and on-chip area. Fig. 8 and Fig. 9 shows two possible configurations of the memristor elements combined into the crossbar that effect the processing speed and number of elements in the system. Fig. 8 illustrates the configuration that ensures the maximum processing speed. The bits representing the pixels are stored in a separate memristors not connected

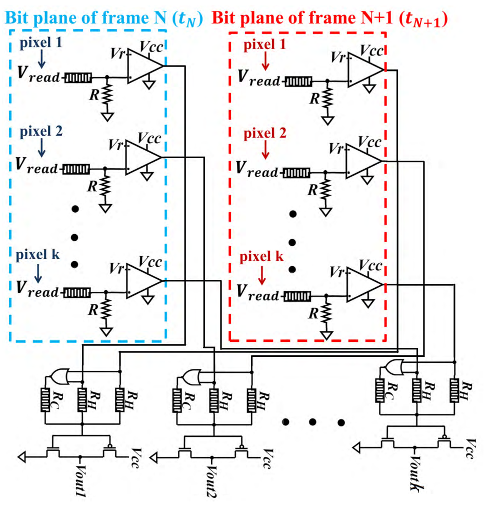

FIGURE 8. Time efficient approach with single memristors are used as storage elements. All the pixels are read in parallel.

to each other. Each memristor is connected to the separate XOR gate. This approach allows to remove the switching process that is used to read the elements when they are connected into the crossbar. This process can be efficient when the number of elements in the process system is small or when the regression of the number of features in the processed visual data is applied. This approach requires high power dissipation and large on-chip area, if the number of processed features is large.

Fig. 9 represents the modular approach. In this approach, the image is divided into several parts and each part is saved in a separate crossbar. This approach allows to achieve the optimum point between the processing time and number of elements in the system. Each crossbar has a separate readout circuit and XOR gate, and all the crossbar are read in parallel. The signal memristive elements within the crossbar are read sequentially. The use of the smaller crossbars is beneficial because the sneak path effects and parasitics effect the output less, comparing to the larger crossbar array. However, complex external or digital control circuit is required.

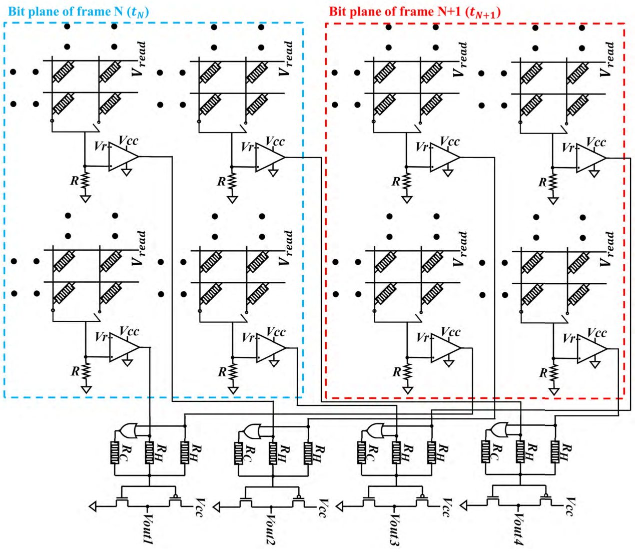
Bit plane of frame N (t_N)
Bit plane of frame N+1 (t_{N+1})
V_{read}
Vr
Vcc
R
R_C
R_H
R_H
Vout1
Vout2
Vout3
Vout4

FIGURE 9. Modular approach. The large image is divided into smaller parts and saved into the separate crossbar. The memristors from a separate crossbars are read in parallel. The memristors withing the crossbar are read sequentially. This approach allows to achieve the desired trade-off depending on the application.

After the comparison, the weighted addition of bit planes is implemented using a summing amplifier to obtain a grayscale image shown in Fig. 10. Weights are represented by the resistance ratios and the MSB bit plane has the largest weight corresponding to the smallest resistance R_{IN} in the summing operational amplifier. The output voltage of the summing amplifier V_{sum} is shown in Eq. 9.

$$V_{sum} = -\left[\frac{R_F}{R_{IN1}}V_1 + \ldots + \frac{R_F}{R_{IN4}}V_4\right], \tag{9}$$

where V_{sum} is the output voltage of the summing amplifier, R_F is the feedback resistance, R_{IN1}, R_{IN2}, R_{IN3}, R_{IN4} are the input resistances, and V_1, V_2, V_3, V_4 are the input voltages.

According to the algorithm, the next step is the thresholding of the obtained grayscale image. This is done using a comparator circuit following the weighted summation operation shown in Fig. 10. Depending on the value of the grayscale pixel, either "0" or "1" is produced at the output V_{out}, and the final binary image is formed according to the Eq.10.

$$|V_{out}| = \begin{cases} 1 & \text{if } |V_{sum}| > V_{ref} \text{ is even} \\ 0 & \text{if } \textit{otherwise} \end{cases} \tag{10}$$

In the final stage, the detected object information is stored in memristive CAMs, as shown in Fig. 11. Memristive CAMs serve several purposes in the circuit and is based on 2T-2M topology with 2 transistors an 2 memristors [36]. Binary data is stored in the CAM array in the form of high or low resistance of the memristors. The charge stored in the row capacitor is leaked through a mismatched cell, where transistor is ON and memristor has a low resistance opening a path to ground. In the match state, when the resistance is

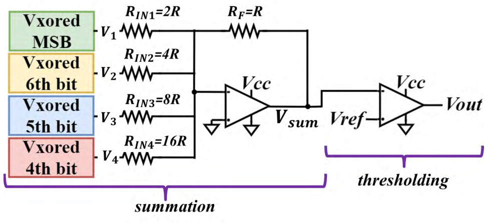

FIGURE 10. Implementation of weighted summation and thresholding stages. Outputs from the XOR TLGs are accepted as inputs to this stage.

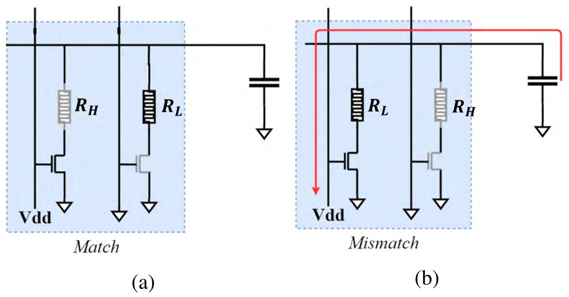

FIGURE 11. 2T-2M CAM cell operation: (a) match case and (b) mismatch case. In the mismatch case, the red line illustrates the discharge path [36].

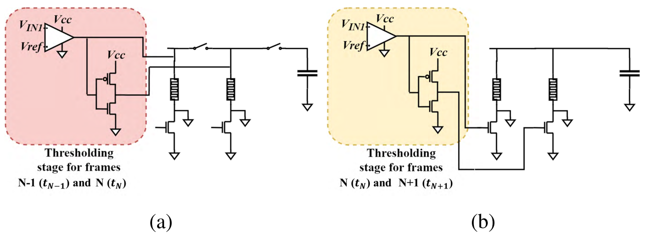
V_{IN1}
Vref
Vcc
Thresholding stage for frames N-1 (t_{N-1}) and N (t_N)
(a)
Thresholding stage for frames N (t_N) and N+1 (t_{N+1})
(b)

FIGURE 12. Write and read cycles for memristive CAM array. (a) write cycle, (b) read cycle.

high and transistor is ON or resistor is low and transistor is OFF, the leakage of the charge to the ground does not occur.

The memristive CAM array operates in two cycles: write cycle and read cycle. The operation cycle are shown in Fig.12. In the first reporting period (write cycle), the outputs from the first pair of compared, combined, and thresholded bit planes of consequent frames are written into the memristors. The transistor inputs, as well as the row capacitors, are disconnected, and the memristors are updated during the write cycle. During the second cycle (read cycle), the outputs from the second pair of compared, combined, and thresholded frames (consequent frames in time) are provided as inputs to the transistor's gate, and a comparison is carried out. Since a complementary output is required, the thresholding stage is embedded with an inverter to produce complementary outputs.

By observing the difference in discharge operation of the capacitor, it is possible to determine the horizontal, vertical, or diagonal direction of object movement. Figure 13 illustrates an example of a horizontally moving object, where only

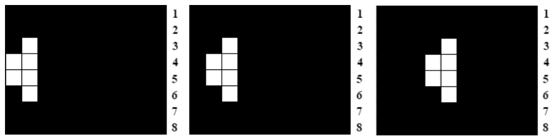

FIGURE 13. Change of pixels depicting an object moving in horizontal direction, with rows indicated by numbers.

the rows 3, 4, 5, 6 demonstrate changes in pixels and the consequent changes in discharge; the rows 1, 2, 7, 8 remain idle. During the object movement, only the rows with changed pixels will have variations in their discharge curves. In the memristive arrays, each memristor corresponds to the storage of a single pixel. When the output from XOR gate of the first two images is compared to the XOR gate output of the last two images, the discharge path of the capacitors in the rows 1, 2, 7, 8 will not change. In the update and write operation, the CAM array enables less memory utilization and less power dissipation by overwriting only the row bits (memristors) that are different and leaving the unchanged bits idle.

IV. EXPERIMENTAL ANALYSES AND RESULTS

The algorithm was analyzed using MATLAB. The algorithm is verified using target videos from the VISOR database [57]. The experiments were conducted on videos recorded under different conditions (outdoor, indoor, etc.). The videos contain various occlusions and illumination changes. In each experiment, 200 video frames have been taken for the analysis. The system level simulation results represent the average recognition accuracy for the performed analysis. The hardware simulations were performed using SPICE. The large number of circuit elements in the crossbar arrays and circuits for image reading, storage, and processing makes the manual writing of SPICE netlist a difficult and time-consuming task. We use MATLAB scripts to generate the SPICE code for simulating the circuits with particular number of memristors in a crossbar, and for automatically analyzing the simulation outputs. The HP memristor model [58], [59] used in the simulations had a large R_{OFF}/R_{ON} ratio of 10^3. This memristor model is suitable for large scale simulations and considers the nonlinear behavior of the memristors. The CMOS technologies used were the TSMC process BSIM models with a feature size of 0.25 μm. To test the performance of the circuit, the power supply for the circuit (V_{CC}) was $1V$, and the logic levels were 1 V for logic high and 0 V for logic low. The inverter was designed to have a threshold voltage of 0.5 V. The same performance can be achieved with the higher power supply (about $3.3V$), which makes possible the integration of the proposed system into the traditional pixel sensors.

To illustrate the efficiency of the proposed method, comparisons with other conventional methods such as the inter-frame differencing (ID), three-frame differencing (TD), background subtraction (BS), and optical flow (OF) methods, were performed, and the results are shown in Table 1 and Fig 14. Quantitative examinations, in terms of accuracy

TABLE 1. Comparisons of performances of the proposed method and the state-of-the-art algorithms.

Measurements	ID [6]	TD [6]	BS [4]	OF [5]	Proposed Method
TRDR	1	1	0.553	1	0.989
FAR	0.527	0.525	0.188	0.523	0
DR	1	1	0.553	1	0.989
Specificity	0	0.019	0.886	0.019	1
Accuracy	0.472	0.479	0.729	0.482	0.995
PP	0.472	0.474	0.813	0.477	1
NP	0	1	0.689	1	0.99
FNR	0	0	0.446	0	0.01
FPR	1	0.98	0.114	0.98	0

and detection rate, were conducted based on ground truth measurements. The frame-based ground truth method was used to compare each frame separately in terms of intensity, position, and quantity of objects, without considering the similarities of the objects in the video sequences. The comparisons were accomplished in terms of the tracker detection rate (TRDR), false alarm rate (FAR), detection rate (DR), specificity, accuracy, positive prediction (PP), negative prediction (NP), false negative rate (FNR), and false positive rate (FPR) metrics [60].

The crossbar array performance was observed by storing four high-order bit planes of three target images, imitating three consequent frames. The images had a size of 8×8 pixels and distinct differences from each other. The bit sequences were extracted from the comparator circuits in the form of voltage waveforms. The circuit simulation results were exported into MATLAB in order to visualize the extracted sequences. The first frame with constituting bit planes is illustrated in Fig. 15, while the bit planes extracted from the crossbar arrays are shown in Fig. 16.

Comparisons of the respective bit planes of successive frames were accomplished using TLGs. The manually XORed bit planes of the first pair of frames are illustrated in Fig. 17, while the compared bit planes obtained from the circuit are depicted in Fig. 18.

In order to observe the operation of the weighted summation and thresholding, the results obtained from the algorithm and from the circuit were illustrated. Manual weighted summation was accomplished by multiplying the respective bit planes with the assigned weights and adding them, while the output from circuit simulation results was exported and read via MATLAB.

The simulated thresholding results were obtained by using a value of 0.25. The circuit output was exported and read. The circuit's reference voltage V_{REF} was also set to $0.25V$. The weighted summation and thresholding outputs are depicted in Fig. 19 and 20, respectively.

The output from the CAM array was formed by observing the discharge curves from the matching and mismatching of the thresholded images from the first and second pairs of

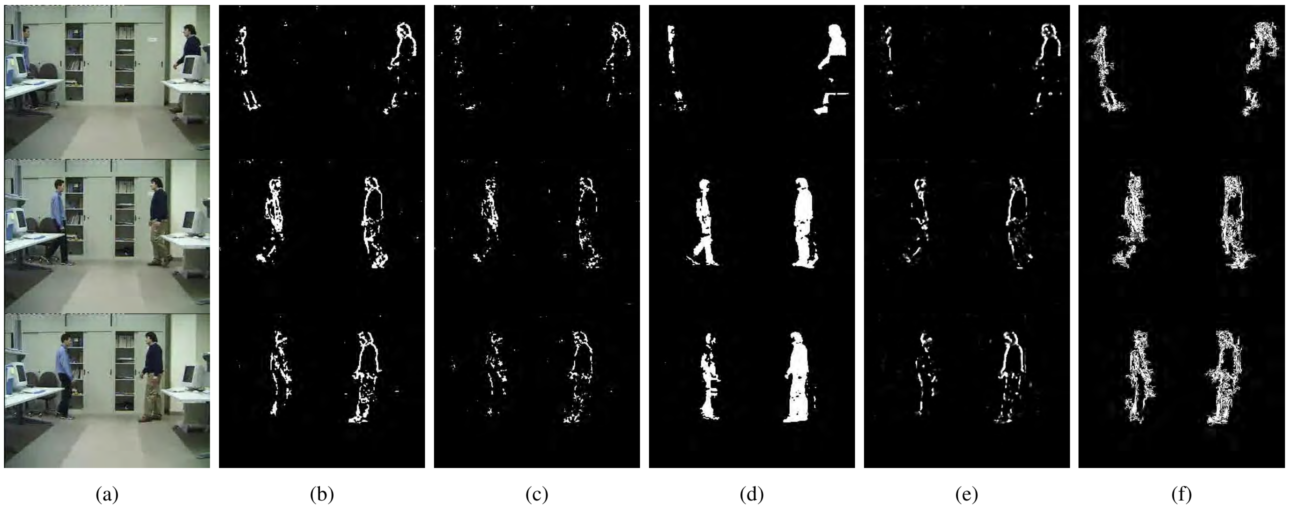

FIGURE 14. Comparisons between proposed moving-object detection method and existing methods, using VISOR database. Column wise: (a) original images, (b) two-frame differencing, (c) three-frame differencing, (d) background subtraction, (e) optical flow, and (f) proposed method. Row-wise, from top down we have, 122nd frame, 147th frame, and 152nd frame.

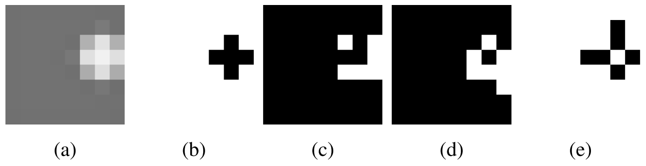

FIGURE 15. Target image along with the extracted bit planes. (a) Original image, (b) MSB plane, (c) 6th bit plane, (d) 5th bit plane, and (e) 4th bit plane.

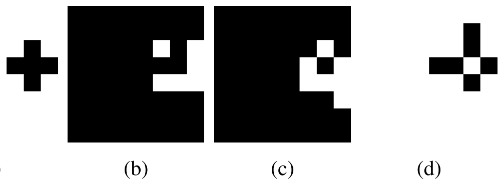

FIGURE 16. Target image bit planes stored and extracted from memristive crossbar array. (a) MSB plane, (b) 6th bit plane, (c) 5th bit plane, and (d) 4th bit plane.

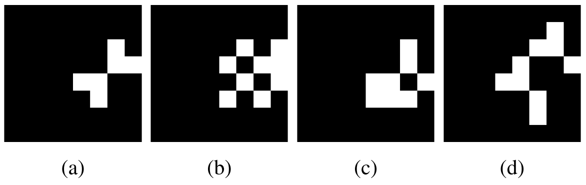

FIGURE 17. Manually XORed bit planes of first pair of consecutive frames. (a) MSB plane, (b) 6th bit plane, (c) 5th bit plane, and (d) 4th bit plane.

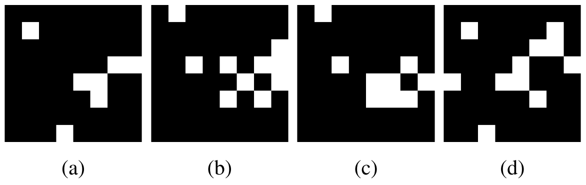

FIGURE 18. XORed bit planes obtained from TLGs. (a) MSB plane, (b) 6th bit plane, (c) 5th bit plane, and (d) 4th bit plane.

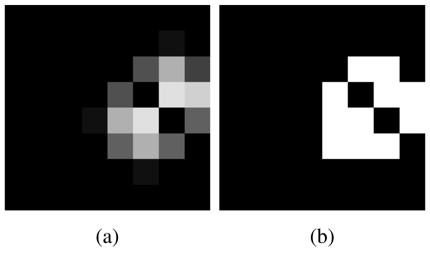

FIGURE 19. Manually added and thresholded bit planes of first pair of consecutive frames. (a) Weighted addition of compared bit planes, (b) thresholded image of (a).

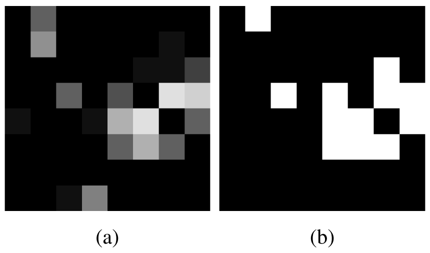

FIGURE 20. Added and thresholded bit planes of first pair of consecutive frames obtained from circuit. (a) Weighted addition of compared bit planes, (b) thresholded image of (a).

frames. The discharge behavior was dependent on the number of different bits in each row. The *second* row matched all bits, the *first* row was mismatched by 1 bit, the *third* row was mismatched by 2 bits, and the *seventh* row was mismatched by 5 bits. The discharge curves from the CAM array are shown in Fig. 21.

Deviations were present among the circuit outputs and processed images, with errors introduced by TLG. Because of the nonlinearity and second-order effects of CMOS technologies

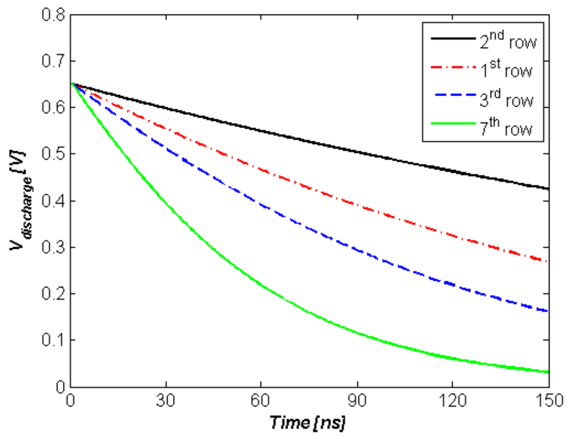

FIGURE 21. Capacitor voltage discharge curves for match and mismatch cases.

TABLE 2. Accuracy, area, power, and delay parameters.

Array size	Accuracy (%)	Area (μm^2)	Power (μW)	Delay (μs)
4x4	75	1398.2	27.34	6.35
8x8	90.6	2703.5	51.63	6.35
16x16	97.3	5932.3	107.3	6.35

and op-amps, a signal delay was introduced at the output of each stage. Since parallel operation was ensured by the crossbar array, the delay in the memristive-CMOS circuits did not depend on the array size, and it did not impact the output results. The accuracy, power, and area of the proposed circuit for three size configurations are shown in Table 2. The power, area and delay calculations are derived from the SPICE simulations. The accuracy results are obtained comparing the circuit simulation results with the original algorithm in Matlab. The area and power shown in Table 2 are calculated for the complete circuit corresponding to the particular crossbar array sizes using the approach mentioned in Fig. 7, when the memristor values are read one at a time. The delay caused by the reading OpAmp in the crossbar, summation and thresholding operation is calculated for a single read cycle. The accuracy was also measured in terms of the temperature sensitivity, as illustrated in Fig. 22. The temperature sensitivity was mostly due to the OpAmps used in the comparison, addition, and thresholding stages, as the memristors' OFF/ON resistance ratios are not affected much by temperature variations [61]. The errors were introduced during the comparison stage. Although the errors were issues for small arrays, the number of incorrectly compared pixels was fixed in the range of 4-6 pixels, which still led to a high accuracy when using a large array size. According to the results, the proposed circuit was also tolerant to temperature fluctuations, and it demonstrated satisfactory operation for temperatures up to 60°C, for small array sizes, and up to 80°C, for larger array sizes.

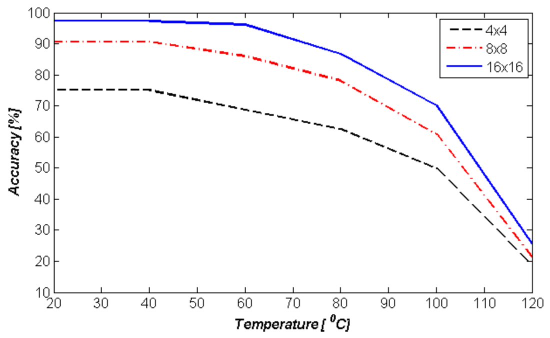

FIGURE 22. Temperature sensitivity graph showing dependence of accuracy on temperature.

TABLE 3. Comparison of CMOS design and memristor-based design of CAM and XOR gates.

Component	CMOS design		Memristive design	
	area	*power*	*area*	*power*
CAM cell	4.000 μm^2	8.880 pW	2.005 μm^2	6.250 pW
XOR gate	3.015 μm^2	30.123 μW	6.000 μm^2	0.551 μW

Table 3 shows the comparison between the memristive design used in the proposed system and equivalent CMOS design of XOR TLG circuit and CAM array cell. The 4T CMOS CAM cell selected for comparison is introduced in [62]. The area calculation does not include the capacitor storing the charge. The CMOS design of XOR TLG used for comparison is a standard CMOS design based on 8 transistors and 2 additional inverters with 2 transistors per each (12 CMOS transistors in total).

V. DISCUSSION

The advantages of the proposed system includes high detection accuracy of the proposed algorithm, the low power dissipation, small on-chip area and scalability of the memristive crossbar array, which enables the possibility of implementing the large system. The use of memristive crossbars to store the visual bit planes, memristive XOR gates and memristive CAM array significantly reduces required on-chip area and power dissipation, comparing to CMOS circuits. According to Table 3 that highlights the advantages of the memristive circuit, comparing to the CMOS design, on-chip area and power dissipation are significantly lower in the memristive design. As in the crossbar approach the writing is performed for one memristor at a time, the overall power required for the memristor update process in the whole system is not high, even if the writing cycle is slow. The slow writing speed is compensated by the fast reading operation from the memristors. In addition, the other advantage of the memristive system over the CMOS design is the scalability of the crossbar with lower leakage current, comparing to the CMOS implementation.

The FPGA solutions for moving object detection have hight power dissipation comparing to the proposed analog hardware based moving object detection module.

For example, one of the recent works shows that the power consumption of FPGA-based module is $0.0026W$ [48]. The mixed signal 64×64 $0.18\mu m$ CMOS image sensor for moving object detection has area and power consumption of $2.25\mu m^2$ and $0.4mW$, respectively [53]. The equivalent 64×64 design based on the proposed circuit with $0.25\mu m$ CMOS process and modular approach (Fig. 9) can consist of the either four separate 16×16 crossbars or eight 8×8 crossbars. The area and power consumption of this circuits are $0.02373\mu m^2$ and $0.429mW$ for 16×16 modular crossbars and $0.02162\mu m^2$ and $0.413\mu W$ for 8×8 modular crossbars, respectively. The power consumption part of the proposed system can be improved by using low power amplifier in the summation part and replacing the resistors with a parallel and series combinations of memristors.

Overall, the existing mixed-signal, FPGA and analog implementations of the moving object detection task have scalability, large on-chip area and high power dissipation problems [48], [53], [55]. Due to small on-chip area and scalability of the memristive circuits, the proposed memristive-CMOS moving object detection module is an appropriate solution to these problems. In addition, with the flexibility of the design of the system and size of the crossbar (shown in Fig. 8 and Fig. 9), it is possible to achieve the trade-off between the processing time, on-chip area and power consumption required for a particular application and integration into the existing system.

Comparing to the other existing memristor-based moving object detection system [56], the system shows higher accuracy for VISOR dataset (99.5%), comparing to 93.3% of the previously introduced method. In addition, the proposed system can be integrated into the traditional CMOS sensor system, comparing to the method proposed in [56], which requires the additional sensing circuit to be implemented.

However, the proposed memristive implementation of the system has several drawbacks and uncertainties that should be investigated in future. For example, comparing to the memristor models, the use of real memristive devices introduces various issues concerning the switching of the memristors, variability of resistance levels, integration of the CMOS devices with memristors in a fabrication process, stability issues, endurance and lifetime of the memristive devices. In the large crossbars, the sneak path problems can not be completely avoided and their effects on the system performance should be investigated further. The impact of the size of the crossbar, leakage currents and parasitics on the accuracy of the system should also be investigated as a part of future work. Frequency and electromagnetic effects should be considered. In addition, the design of the operational amplifier (OpAmp) used in the CMOS adder and comparator can be improved or replaced with low power OpAmp to achieve lower power dissipation and smaller area.

VI. CONCLUSION

In this paper, we presented a moving-object detection algorithm implemented using a memristor crossbar array architecture that could store and process the images in bit planes. The respective bit planes of consecutive frames were compared using memristive TLGs, and grayscale images were generated by weighted addition. The binary image of the detected object was obtained by setting an appropriate thresholding level. The last stage comprised a CAM array, which was used for two purposes: as a long-term memory and for providing analysis capability. The crossbar arrangement and low-level vision processing abilities made it a appropriate for integration with the existing image sensors and perform near sensor processing and edge computing tasks. The sneak path problem could be avoided by reading one element at a time; however, this led to significant reading delays for larger-sized arrays. As a future work, the TLG operations can be improved for achieving better accuracy.

REFERENCES

[1] S. Manchanda and S. Sharma, "Analysis of computer vision based techniques for motion detection," in *Proc. 6th Int. Conf.-Cloud Syst. Big Data Eng. (Confluence)*, Jan. 2016, pp. 445–450.

[2] Y. Liu and M. Zhi, "Moving object detection in tennis video," in *Proc. 8th Int. Conf. Intell. Netw. Intell. Syst. (ICINIS)*, Nov. 2015, pp. 109–112.

[3] B. Sugandi, H. Kim, J. K. Tan, and S. Ishikawa, "Tracking of multiple moving objects under outdoor environment using color-based particle filter," in *Proc. 3rd IEEE Int. Conf. Comput. Sci. Inf. Technol. (ICCSIT)*, vol. 1. Jul. 2010, pp. 103–107.

[4] S. Manchanda and S. Sharma, "Identifying moving objects in a video using modified background subtraction and optical flow method," in *Proc. 3rd Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, Mar. 2016, pp. 129–133.

[5] J. S. Kulchandani and K. J. Dangarwala, "Moving object detection: Review of recent research trends," in *Proc. Int. Conf. Pervasive Comput. (ICPC)*, Jan. 2015, pp. 1–5.

[6] I. Kartika and S. S. Mohamed, "Frame differencing with post-processing techniques for moving object detection in outdoor environment," in *Proc. IEEE 7th Int. Colloq. Signal Process. Appl. (CSPA)*, Mar. 2011, pp. 172–176.

[7] H. Mehrotra, G. S. Badrinath, B. Majhi, and P. Gupta, "Bit plane slicing technique for iris based biometric system," in *Proc. 1st Int. Conf. Parallel Distrib. Grid Comput. (PDGC)*, Oct. 2010, pp. 350–355.

[8] C.-Y. Lin, Z.-Y. Jian, and W.-Y. Lin, "Image bit-planes representation for moving object detection in real-time video surveillance," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, May 2016, pp. 1–2.

[9] L. Lei, S. W. Kim, W. J. Park, D. H. Kim, and S. J. Ko, "Eigen directional bit-planes for robust face recognition," *IEEE Trans. Consum. Electron.*, vol. 60, no. 4, pp. 702–709, Nov. 2014.

[10] B. Bonney, R. Ives, D. Etter, and Y. Du, "Iris pattern extraction using bit planes and standard deviations," in *Proc. Conf. Rec. 38th Asilomar Conf. Signals, Syst. Comput.*, vol. 1. Nov. 2004, pp. 582–586.

[11] T. Z. Lee and D. B. L. Bong, "Face and palmprint multimodal biometric system based on bit-plane decomposition approach," in *Proc. IEEE Int. Conf. Consum. Electron.-Taiwan (ICCE-TW)*, May 2016, pp. 1–2.

[12] B. F. Wu, C. C. Chiu, and Y. L. Chen, "Algorithms for compressing compound document images with large text/background overlap," *IEE Proc.-Vis., Image Signal Process.*, vol. 151, no. 6, pp. 453–459, Dec. 2004.

[13] S. R. Dubey, S. K. Singh, and R. K. Singh, "Local bit-plane decoded pattern: A novel feature descriptor for biomedical image retrieval," *IEEE J. Biomed. Health Inform.*, vol. 20, no. 4, pp. 1139–1147, Jul. 2016.

[14] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, pp. 80–83, May 2008.

[15] T. Prodromakis and C. Toumazou, "A review on memristive devices and applications," in *Proc. 17th IEEE Int. Conf. Electron., Circuits, Syst. (ICECS)*, Dec. 2010, pp. 934–937.

[16] S. Hamdioui, H. Aziza, and G. C. Sirakoulis, "Memristor based memories: Technology, design and test," in *Proc. 9th IEEE Int. Conf. Design Technol. Integr. Syst. Nanoscale Era (DTIS)*, Santorini Island, Greece, May 2014, pp. 1–7.

[17] W. Chen, W. Lu, Y. Li, K. Alexander, and R. Jha, "An integrated active-pixel-sensor and memristive platform for neural-inspired image learning and recognition," in *Proc. IEEE 57th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2014, pp. 741–744.

[18] A. K. Maan and A. P. James, "Voltage controlled memristor threshold logic gates," in *Proc. IEEE Asia Pacific Conf. Circuits Syst. (IEEE APCCAS)*, Oct. 2016, pp. 376–379.

[19] I. Vourkas and G. C. Sirakoulis, "Emerging memristor-based logic circuit design approaches: A review," *IEEE IEEE Circuits Syst. Mag.*, vol. 16, no. 3, pp. 15–30, 3rd Quart., 2016.

[20] A. H. Edwards, H. J. Barnaby, K. A. Campbell, M. N. Kozicki, W. Liu, and M. J. Marinella, "Reconfigurable memristive device technologies," *Proc. IEEE*, vol. 103, no. 7, pp. 1004–1033, Jul. 2015.

[21] J. Yang, Y.-D. Sun, M.-J. Wu, and Q.-N. Zhang, "Multi-class moving target detection with Gaussian mixture part based model," in *Proc. IEEE Int. Conf. Consumer Electron. (ICCE)*, Jan. 2014, pp. 386–387.

[22] A. Sobral and A. Vacavant, "A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos," *Comput. Vis. Image Understand.*, vol. 122, pp. 4–21, May 2014.

[23] W. Kim and C. Jung, "Illumination-invariant background subtraction: Comparative review, models, and prospects," *IEEE Access*, vol. 5, pp. 8369–8384, 2017.

[24] Y. Lin, M. Fang, and D. Shihong, "An object reconstruction algorithm for moving vehicle detection based on three-frame differencing," in *Proc. IEEE 12th Int. Conf. Ubiquitous Intell. Comput., IEEE 12th Int. Conf. Auto. Trusted Comput., IEEE 15th Int. Conf. Scalable Comput. Commun. Associated Workshops (UIC-ATC-ScalCom)*, Aug. 2015, pp. 1864–1868.

[25] M. I. lsever and C. Ünsalan, "Pixel-based change detection methods," in *Two-Dimensional Change Detection Methods*. London, U.K.: Springer-Verlag, 2012, pp. 7–21.

[26] D. Lu, P. Mausel, E. Brondízio, and E. Moran, "Change detection techniques," *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2365–2401, 2004.

[27] G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic*, vol. 4. Englewood Cliffs, NJ, USA: Prentice-Hall, 1995.

[28] T. Raja and S. Mourad, "Digital logic implementation in memristor-based crossbars," in *Proc. Int. Conf. Commun., Circuits Syst.*, Jul. 2009, pp. 939–943.

[29] R. Rosezin, E. Linn, L. Nielen, C. Kügeler, R. Bruchhaus, and R. Waser, "Integrated complementary resistive switches for passive high-density nanocrossbar arrays," *IEEE Electron Device Lett.*, vol. 32, no. 2, pp. 191–193, Feb. 2011.

[30] C.-M. Jung, J.-M. Choi, and K.-S. Min, "Two-step write scheme for reducing sneak-path leakage in complementary memristor array," *IEEE Trans. Nanotechnol.*, vol. 11, no. 3, pp. 611–618, May 2012.

[31] S. Kannan, J. Rajendran, R. Karri, and O. Sinanoglu, "Sneak-path testing of crossbar-based nonvolatile random access memories," *IEEE Trans. Nanotechnol.*, vol. 12, no. 3, pp. 413–426, May 2013.

[32] A. K. Maan, D. A. Jayadevi, and A. P. James, "A survey of memristive threshold logic circuits," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 8, pp. 1734–1746, Aug. 2016.

[33] M. Teimoori, A. Ahmadi, S. Alirezaee, S. V. A.-D. Makki, and M. Ahmadi, "A novel memristor based integrate-and-fire neuron implementation using material implication logic," in *Proc. IEEE 28th Can. Conf. Electr. Comput. Eng. (CCECE)*, May 2015, pp. 1176–1179.

[34] C. B. Dara, T. Haniotakis, and S. Tragoudas, "Low power and high speed current-mode memristor-based TLGs," in *Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFT)*, Oct. 2013, pp. 89–94.

[35] K. Pagiamtzis and A. Sheikholeslami, "Content-addressable memory (CAM) circuits and architectures: A tutorial and survey," *IEEE J. Solid-State Circuits*, vol. 41, no. 3, pp. 712–727, Mar. 2006.

[36] M. A. Bahloul *et al.*, "Design and analysis of 2T-2M ternary content addressable memories," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1430–1433.

[37] X. Zhang, C. Zhu, S. Wang, Y. Liu, and M. Ye, "A Bayesian approach to camouflaged moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 9, pp. 2001–2013, Sep. 2017.

[38] A. Keivani, J.-R. Tapamo, and F. Ghayoor, "Motion-based moving object detection and tracking using automatic k-means," in *Proc. IEEE AFRICON*, Sep. 2017, pp. 32–37.

[39] J. Cao, X. Sun, S. Zhao, Y. Wang, and S. Gong, "Algorithm of moving object detection based on multifeature fusion," in *Proc. IEEE Int. Conf. Inf. Autom. (ICIA)*, Jul. 2017, pp. 931–935.

[40] Z. Zhou, P. Liu, G. Chen, and Y. Liu, "Moving object detection based on zernike moments," in *Proc. 5th Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT)*, Dec. 2016, pp. 696–699.

[41] M. Shakeri and H. Zhang, "Moving object detection in time-lapse or motion trigger image sequences using low-rank and invariant sparse decomposition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5133–5141.

[42] Z. Xiao, B. Dai, T. Wu, L. Xiao, and T. Chen, "Dense scene flow based coarse-to-fine rigid moving object detection for autonomous vehicle," *IEEE Access*, vol. 5, pp. 23492–23501, 2017.

[43] Y. Chen, J. Wang, B. Zhu, M. Tang, and H. Lu, "Pixel-wise deep sequence learning for moving object detection," *IEEE Trans. Circuits Syst. Video Technol.*, to be published, doi: 10.1109/TCSVT.2017.2770319.

[44] L. Zhu, Y. Hao, and Y. Song, "$L_{1/2}$ norm and spatial continuity regularized low-rank approximation for moving object detection in dynamic background," *IEEE Signal Process. Lett.*, vol. 25, no. 1, pp. 15–19, Jan. 2018.

[45] X. Cao, L. Yang, and X. Guo, "Total variation regularized RPCA for irregularly moving object detection under dynamic background," *IEEE Trans. Cybern.*, vol. 46, no. 4, pp. 1014–1027, Apr. 2016.

[46] Y. Wu, X. He, and T. Q. Nguyen, "Moving object detection with a freely moving camera via background motion subtraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 2, pp. 236–248, Feb. 2017.

[47] S. Chen, T. Xu, D. Li, J. Zhang, and S. Jiang, "Moving object detection using scanning camera on a high-precision intelligent holder," *Sensors*, vol. 16, no. 10, p. 1758, 2016.

[48] H.-Y. Chen and Y.-K. Wang, "Hardware design of moving object detection on reconfigurable system," *J. Comput. Commun.*, vol. 4, no. 10, p. 30, 2016.

[49] S. Cherian, C. S. Singh, and M. M, "Implementation of real time moving object detection using background subtraction in FPGA," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2014, pp. 867–871.

[50] S. Sajjanar, S. K. Mankani, P. R. Dongrekar, N. S. Kumar, Mohana, and H. V. R. Aradhya, "Implementation of real time moving object detection and tracking on fpga for video surveillance applications," in *Proc. IEEE Distrib. Comput., VLSI, Electr. Circuits Robot. (DISCOVER)*, Aug. 2016, pp. 289–295. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7806248

[51] W. Liu, H. Chen, and L. Ma, "Moving object detection and tracking based on ZYNQ FPGA and ARM SOC," in *Proc. IET Int. Radar Conf.*, Oct. 2015, pp. 1–4.

[52] K. Nishio, Y. Miyauchi, and S. Ideta, "Simple motion detection circuits and system for mobile robot based on biological vision system," in *Proc. 13th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2014, pp. 65–69.

[53] B. Zhao, X. Zhang, and S. Chen, "A CMOS image sensor with on-chip motion detection and object localization," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2011, pp. 1–4.

[54] K. Nishio, H. Yonezu, S. Sawa, Y. Yoshikawa, and Y. Furukawa, "Analog integrated circuit for detection of approaching object against moving background based on lower animal vision," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 4. May 2004, pp. IV-832–IV-835.

[55] M. Kawaguchi, T. Jimbo, and M. Umeno, "Analog VLSI layout design of advanced image processing for artificial vision model," in *Proc. IEEE Int. Symp. Ind. Electron. (ISIE)*, vol. 3. Jun. 2005, pp. 1239–1244.

[56] A. K. Maan, D. S. Kumar, S. Sugathan, and A. P. James, "Memristive threshold logic circuit design of fast moving object detection," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 10, pp. 2337–2341, Oct. 2015.

[57] R. Vezzani and R. Cucchiara, "Video surveillance online repository (ViSOR): An integrated framework," *Multimedia Tools Appl.*, vol. 50, no. 2, pp. 359–380, Nov. 2010, doi: http://dx.doi.org/10.1007/s11042-009-0402-9

[58] Z. Biolek, D. Biolek, and V. Biolková, "SPICE model of memristor with nonlinear dopant drif," *Radioengineering*, vol. 18, no. 2, pp. 210–214, Mar. 2009.

[59] D. Biolek, Z. Kolka, V. Biolková, and Z. Biolek, "Memristor models for spice simulation of extremely large memristive networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 389–392.

[60] V. Manohar, P. Soundararajan, H. Raju, D. Goldgof, R. Kasturi, and J. Garofolo, "Performance evaluation of object detection and tracking in video," in *Proc. Asian Conf. Comput. Vis.*, Jan. 2006, pp. 151–161.

[61] H. Abunahla, B. Mohammad, D. Homouz, and C. J. Okelly, "Modeling valance change memristor device: Oxide thickness, material type, and temperature effects," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 12, pp. 2139–2148, Dec. 2016.

[62] S. Rajendar and P. Ramakrishna, "A novel high performance design of memory architecture using modified 4T CAM cell," in *Proc. Int. Conf. Inventive Commun. Comput. Technol. (ICICCT)*, Mar. 2017, pp. 272–277.

NAZGUL DASTANOVA received the B.S. degree from the Almaty University of Power Engineering and Telecommunications, Almaty, Kazakhstan, in 2015, and the M.S. degree from the School of Engineering, Nazarbayev University, Astana, Kazakhstan, in 2017. She is currently with the embedded systems industry as a Research Staff. Her research interests include the memristor-based real-time image processing.

SULTAN DUISENBAY received the bachelor's and master's degree in electrical and electronic engineering from Nazarbayev University, Kazakhstan, in 2015 and 2017, respectively. He is currently a Staff Member in GE. He is also conducting research in the area of neuromorphic CAM array architectures utilizing memristors. His focus lies in the simulation of various memristor models and their incorporation into CAM array circuits.

OLGA KRESTINSKAYA (GS'16) received the B.E. degree (Hons.) in electrical engineering, with a focus on bio-inspired memory arrays, in 2016. She is currently involved in her graduate degree thesis on neuromorphic memristive system from the Electrical Engineering Department, Nazarbayev University. Her current research interests include hierarchical temporal memory and pattern recognition algorithms.

ALEX PAPPACHEN JAMES (SM'13) received the Ph.D. degree from the Griffith School of Engineering, Griffith University. He has experience of managing industry and academic projects in board design and pattern recognition circuits, and data and business analytics consulting for IT and semiconductor industry. He is currently the Chair of the Electrical Engineering Department, Nazarbayev University, and the IEEE Kazakhstan subsection. He is also involved in brain-inspired circuits, memristor circuits, and algorithms and systems. He is a Senior Fellow of HEA. He is an Associate Editor of *Human-Centric Computing and Information Sciences*, the IEEE ACCESS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS 1, the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, and IET CPS journals.