A Low Power Multi-mode CMOS Image Sensor with Integrated On-chip Motion Detection

Xilin Liu, Milin Zhang, Jan Van der Spiegel
Department of Electrical and Systems Engineering (ESE)
University of Pennsylvania, Philadelphia, PA 19104

***Abstract*—In this paper, we propose a novel low power multi-mode CMOS smart image sensor node with integrated focal-plane motion detection and video compression. An 80×80 image pixel array is fabricated in 0.5 μm 3M2P standard CMOS technology, occupying 3×3 mm^2 silicon area. The proposed imager enables various operational modes, including 1) event generator mode, 2) motion tracking mode and 3) video output mode in full-resolution or compression by region of interest (ROI). An ultra low power focal-plane motion detection block, consisting of analog memory and dual-threshold comparator, is integrated in the pixel-level circuit for on-chip motion detection. A hardware-friendly motion tracking algorithm is developed that indicates ROIs according to a strategy based on the detection results. A 12-bit on-chip, off-array ADC is employed to convert the captured light intensity into digital readouts. In order to further reduce the power consumption, lower image resolution is used under the first two modes. A trade-off analysis between the image resolution and detection accuracy is proposed in this paper. In simulation, the total power consumption is 10μW at a frame rate of 30fps and a supply voltage of 3.3V in motion tracking mode. A compression ratio of 14% and an average PSNR of 42dB is achieved in compressive video output mode.**

***Index Terms*—CMOS image sensor, multi-mode imager, low power, pixel-level motion detector, compressive image acquisition**

I. Introduction

According to the reports from the U.S. Department of Energy[1], residential and commercial building consumes nearly 40% of the total national energy. Therefore, the design and production of new energy efficient technologies are crucial to meet goals such as the Zero Net Energy (ZNE) Building. Building automation system (BAS) seeks the answer to this problem by employing intelligent distributed sensors and actuators[2]. Among various sensors, a smart image sensor is one of the most efficient and powerful detectors for monitoring the environment. Due to the development of CMOS technology, the market share of CMOS image sensor (CIS) has been increasing in the last two decades. CIS enables the integration of image capture and image processing into a single die, realizing the concept of camera-on-chip [3]-[8].

For a smart sensor implemented in a BAS, the capabilities of feature extraction and motion tracking are important. Traditional motion tracking algorithms can be classified into: 1) temporal difference based method, which calculates the changes of the pixel intensities in continuous frames[3]; 2) correlation based method, which calculates the product of local pixel intensity and the delayed intensity at neighboring location[4]; 3) gradient based method, which calculates the ratio of temporal and spatial derivatives of the intensity[5]; 4) token based method, which extracts and tracks particular features, such as edges or zero-crossings[6]; and 5) cluster based method, which divides motion events into clusters based on certain criterion[7]. In this paper, a hardware friendly temporal difference based motion tracking algorithm is proposed. The proposed algorithm is implemented in a pixel-level circuit. Mathematical analysis is used to solve the trade-off between tracking accuracy and algorithm complexity. Variable resolution, internal pixel-level memory and dual-threshold comparator are employed to reduce circuit complexity without sacrificing too much accuracy.

In addition to motion tracking and video capture, another useful function for BAS is also realized in the proposed design, which is the compressive video acquisition based on ROI. The ROIs are highlighted according to motion detection results. Different refresh rates are employed for the ROIs and background. The background is modeled off-chip to be able to distinguish fake ROIs from the real objects of interest.

The paper is organized as follows. Section II presents an overview of the entire system. Different operation modes and algorithms are introduced in this section. Section III proposes the system architecture, circuit implementation and simulation results. While Section IV concludes the whole paper.

II. System Overview

A block diagram of the entire system is illustrated in Fig.1. The proposed low power multi-mode CMOS imager enables various modes of operation: I) event generation mode,

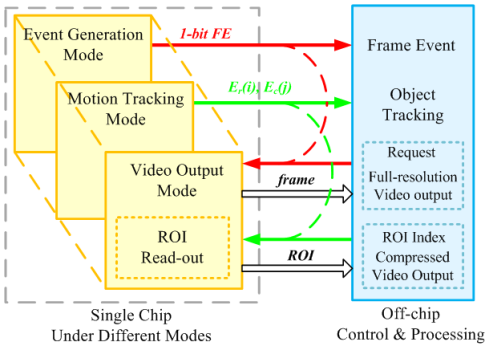

Fig. 1. System block diagram of the proposed low power multi-mode CMOS smart image sensor. On-chip imager can be configured as I) event generation mode, II) motion tracking mode, and III) video output mode.

which generates 1-bit Frame-Event (FE) signal once motion or flashing happens in the focal plane; II) motion tracking mode, which tracks the moving object based on temporal changes in focal plane; and III) video output mode, which is capable of capturing continuous frames at full-resolution or in the ROI only. Video output mode can be set to wake up when triggered by the FE generated in mode I. The ROI is tracked by the imager based on a proposed strategy. Higher refresh rate is employed for ROIs than background in mode III. Background is modeled off-chip by Gaussian Mixture Model (GMM) to further distinguish the foreground.

A. Event Generation Mode

Assume $I_{t[k]}$ is the intensity matrix of the frame captured at $t = t[k]$, and $I_{t[k-1]}$ is the one captured at $t = t[k-1]$. The differential image is calculated as

$$\Delta I_{t[k,k-1]} = |I_{t[k]} - I_{t[k-1]}| \tag{1}$$

Pixel-event E_p is defined as

$$E_p(i,j) = \begin{cases} 1 & \Delta I_{t[k,k-1]}(i,j) \geq \xi \\ 0 & \Delta I_{t[k,k-1]}(i,j) < \xi \end{cases} \tag{2}$$

where, i, j are the coordinates of the pixel, ξ is a threshold value depending on the noise level. Assume the probability of a pixel-event equals to one is

$$P(E_p(i,j) = 1) = p \tag{3}$$

A Frame-Event is defined as a significant number of pixel-events happening in two consecutive image frames. In an $n \times n$ array, a Frame-Event is defined as

$$FE = \begin{cases} 1 & \sum_{j=1}^{n} \sum_{i=1}^{n} E_p(i,j) \geq \alpha n^2 \\ 0 & \sum_{j=1}^{n} \sum_{i=1}^{n} E_p(i,j) < \alpha n^2 \end{cases} \tag{4}$$

where, α is the threshold for the sum of pixel-events within one frame. However, the workload of counting pixel-events for entire array increases dramatically while larger array is employed. In order to reduce the computation requirement, we introduce row-event E_r and column-event E_c into the detection algorithm.

$$E_r(i) = \begin{cases} 1 & \sum_{j=1}^{n} E_p(i,j) > 0 \\ 0 & \sum_{j=1}^{n} E_p(i,j) = 0 \end{cases} \tag{5}$$

$$E_c(j) = \begin{cases} 1 & \sum_{i=1}^{n} E_p(i,j) > 0 \\ 0 & \sum_{i=1}^{n} E_p(i,j) = 0 \end{cases} \tag{6}$$

An alternative way to generate a Frame-Event (FE') is to use the statistical results of the row and column-events.

$$FE' = \begin{cases} 1 & \sum_{i=1}^{n} E_r(i) + \sum_{j=1}^{n} E_c(j) \geq 2\alpha' n \\ 0 & \sum_{i=1}^{n} E_r(i) + \sum_{j=1}^{n} E_c(j) < 2\alpha' n \end{cases} \tag{7}$$

where, α' is the threshold for the sum of row and column-events. Using the sum of row and column-events instead of pixel-events reduces the workload as well as power consumption from n^2 to 2n. However, the reduction of statistical workload leads to accuracy loss. Assume in an n $\times$ n image array, there are a row/column-events and b column/row-events, while $a > b, b = \beta \cdot a$,where $\beta \in (0,1)$. The probability for m pixel-events happening can be expressed as

$$P(\sum_{j=1}^{n} \sum_{i=1}^{n} E_p(i,j) = m) = C_{ab-a}^{m-a} p^{m-a} (1-p)^{ab-m} \tag{8}$$

A maximum $P(m)$ is achieved when m is the closest integer above $abp - ap + a + p - 1$. Then α' can be optimized by

$$\alpha' = \frac{1+\beta}{4p\beta n}(\sqrt{(1-p)^2 - 4p\beta(p-1-\alpha n^2)} + p - 1) \tag{9}$$

The row and column-event threshold can be determined by this strategy. Simulation confirms that this strategy gives a high accuracy while reducing considerable amount of power and time.

B. Motion Tracking Mode

The motion tracking algorithm is illustrated in Fig.2. The

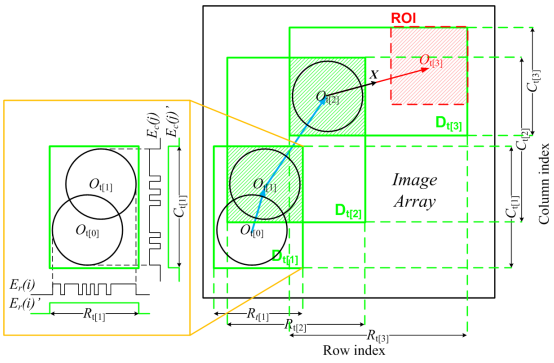

Fig. 2. Motion tracking algorithm description. Circles on the image array represent the trajectory of one moving object in four consecutive frames from $t[0]$ to $t[3]$. The boxed region shows how the motion region is detected. The object's position at the frame of $t[1]$ is estimated by $D_{t[1]} \cap D_{t[2]}$. The prediction of ROI at the frame of $t[3]$ is marked by dashed square.

four circles on the image array represent one moving object in four consecutive frames. The boxed region shows how motion region is detected. As the object moves, the temporal intensity changes in focal plane generate row and column-events E_r and E_c, respectively. Set $R_{t[k]}$/$C_{t[k]}$ contains rows/columns which $E_r(i)$ / $E_c(j)$ equals 1. The motion region $D_{t[k]}$ can be expressed as $R_{t[k]} \cap C_{t[k]}$. However, in practice, the $R_{t[k]}$ and $C_{t[k]}$ are not always continuous within the real object region especially when the object and background are of similar intensity. Thus a single object may cause several motion regions. Gaussian smooth filtering is introduced to perform preprocessing on $R_{t[k]}$ and $C_{t[k]}$. The Gaussian smooth filter modifies the input signal by a convolution with a Gaussian function, which is widely used to reduce image noise. In this paper, we employ it to filter out the sets of row and column-events, resulting in $R'_{t[k]}$ and $C'_{t[k]}$. The hardware friendly 1-D filtering gives a much more accurate motion detection result. The object of the frame at $t = t[k]$ can be determined by:

$$Obj_{t[k]} = D_{t[k]} \cap D_{t[k+1]} \tag{10}$$

Multiple objects in the focal plane can also be tracked using this algorithm, but multiple objects recorded by row

and column events will lead to fake objects in processing. Distinguishing of fake objects and real objects is discussed in the following session.

C. Video Output Mode

The proposed imager enables full-resolution video output as well as compressive video output based on ROI. The center of the ROI of $t = t[k]$ is estimated by

$$\overrightarrow{O_{t[k-1]}O_{t[k]}} = 2 \cdot \overrightarrow{O_{t[k-1]}X_{t[k]}} \tag{11}$$

where $X_{t[k]}$ is the center of $D_{t[k]}$. The size of the ROI is estimated to be the same as the object in the previous frame.

Gaussian mixture model is used to model the background. After the read-out of the ROIs, an evaluation of the ROI is conducted to tell whether it is a real moving object or belongs to the background. Fig.3 shows the simulation result.

(a) (b) (c) (d)

Fig. 3. Simulation of the proposed algorithm. (a) the original frame; (b) the motion tracking results before filtering; (c) the motion detection regions after Gaussian smooth filter is employed, where objects are merged into complete ones; (d) the compressive video output after ROIs evaluated with background.

III. Circuit Implementation And Simulation

A. System Architecture

The proposed CMOS image sensor includes an 80×80 image array, which is divided into 2×2 pixels blocks; row and column processing units control the timing; the event-generator generates 1-bit FE under mode I; row and column buffers latch the motion detection results under mode II; an off-array 12-bit pipeline ADC converts the analog output into digital values under mode III.

Intensity of each pixel is read out serially through a rolling shutter sequence. Delta-difference sampling is performed off-chip to reduce fixed pattern noise (FPN). Motion detection results are read out through a global shutter. Each motion detection block generates a 1-bit digital signal indicating a pixel-event, which directly triggers the row and column-events. Then the row and column-events are read-out through buffer chains. All processing chains are shared in above modes under different configurations.

B. Image Sensor Array

Four standard active pixel sensors (APS) [8] are integrated within each block, as shown in Fig.4. Each APS contains a photodiode, two reset transistors (M1, M2), one source follower (M3), and switches for the output. The intensity read-out process begins from a reset phase which pulls up the photodiode voltage to VDD. After the reset phase, the photodiode voltage decreases as the photon-generated charges accumulate on the photodiode capacitance. The integration voltage is readout and digitized by the off-array ADC during the readout phase.

A motion detection unit is shared by four pixels in each block. Sub-array operation is realized by configuring reset chain. The motion detection unit consists of an analog memory (capacitor) and a pixel-level dual-threshold comparator to generate 1-bit pixel-event E_p defined in Eq. (1-2). The timing of the motion detection mode is shown in Fig.4. During $\phi 1$, both $S1$ and $S2$ are closed for sampling the integrated voltage $V_{t[k-1]}$ with respect to the reference voltage:

$$Q_1 = C(V_{t[k-1]} - V_{ref}) - C_{gb}V_{ref} \tag{12}$$

During $\phi 2$, closing $S1$ and opening $S2$ allow sampling the different voltage between two frames.

$$Q_2 = C(V_{t[k]} - V_{comp}) - C_{gb}V_{comp} \tag{13}$$

where, V_{comp} is the voltage on the input gate of the comparator, C_{gb} is the parasitic capacitor of the input gate. According to the law of charge conservation, $Q_1 = Q_2$

$$V_{comp} = \frac{C}{C + C_{gb}}(V_{t[k]} - V_{t[k-1]}) + V_{ref} \tag{14}$$

By designing the C so C_{gb}'s changing with input voltage is neglectable with respect to C, the V_{comp} can be expressed as

$$V_{comp} = \alpha \cdot \Delta V_{t[k-1,k]} + \beta \cdot V_{ref} \tag{15}$$

where, α, β are constants. The charges on the capacitor are reset during $\phi 3$.

As the input voltage V_{comp} increases, the drain current through M6, M8 and M10 first increases exponentially, and then decreases. M9 and M11 are used to mirror the current to the output branch in serial with a current source M7[9]. The V_{out} depends on the relative current magnitude of mirrored branch current and the current source. Dual-way threshold voltages can be adjusted by biasing M7. Fig.5 shows the simulation result, while the input V_{comp} increases linearly across dual-threshold voltages.

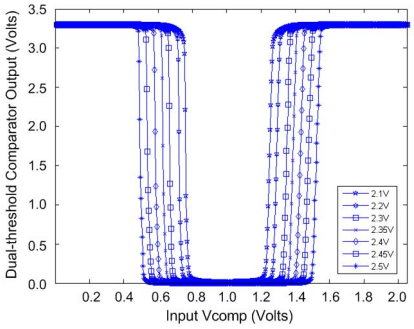

Fig. 5. Simulation results of output when sweeping input voltage linearly under different current source biasing

Eq. (2) is realized in hardware by setting the center voltage of the dual-threshold comparator equal to βV_{ref}. The comparison is finished in only one phase. Dynamic logic is used to generate row and column requests according to Eq. (5-6).

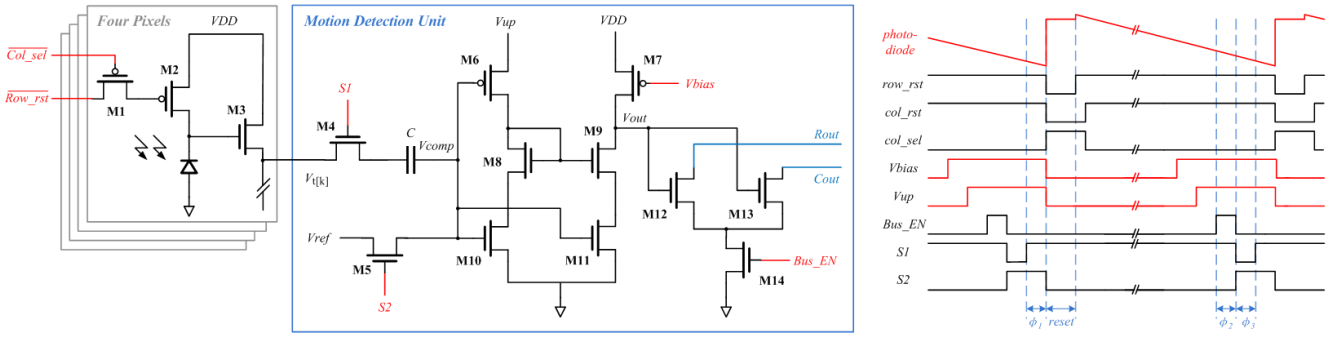

Fig. 4. Proposed motion-detect block circuit and timing. Motion detection unit is shared by four pixels in each block.

Before $\phi 2$, both row and column request buses are charged to a high level. When Bus_{EN} signal is valid, V_{out} of the comparator allows M12 and M13 in each block to discharge the buses to low level in the case of pixel-event happening, or disconnect from the buses.

C. Simulation

A circuit level simulation is performed using SPICE models, as shown in Fig.6. Current source arrays are created by mapping the original image sequences. The power consumption of the imager is 10 μW at a frame rate of 30fps and a supply voltage of 3.3V. Simulation results show a compression ratio of 14% is achievable in BAS implementation and the reconstructed videos show an average PSNR of 42dB.

(a) (b) (c)

Fig. 6. A circuit level simulation using SPICE. (a) original video frame. (b) pixel-level motion detection result. (c) compressive video output based on ROIs.

IV. Conclusion

In this paper, we present a low power multi-mode CMOS image sensor with on-chip motion detection. A hardware-friendly motion tracking algorithm is developed to generate frame-event, row and column-event, and also highlight ROI for compressive video read-out. Trade-offs between imager performance and system cost are performed, including power consumption, circuit complexity, and data bandwidth. Low power designs are realized at both the system and circuit levels. A prototype chip was implemented in 0.5μm standard 3M2P CMOS technology, occupying a silicon area of 3 $\times$ 3 mm^2. The layout is shown in Fig.7. Table I summarizes the characteristics of the proposed design.

Acknowledgment

The authors would like to thank MOSIS Education Program for fabricating the chip.

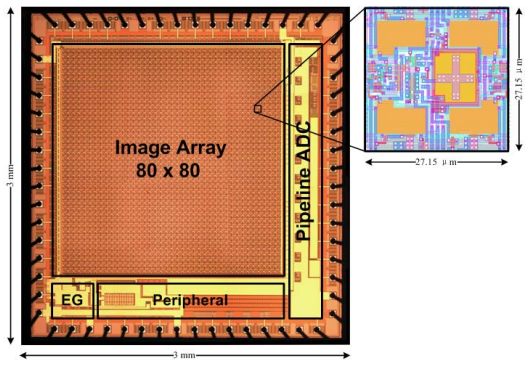

Fig. 7. Microphotography of the fabricated chip and layout of the proposed motion detection block

TABLE I
Chip Parameters Summary

Process Technology	0.5μm CMOS Technology
Die Size	3mm $\times$ 3mm
Pixel Array	80 $\times$ 80
Pixel Size	27.15 $\mu m\times$ 27.15 μm
Fill Factor	26.8%
Motion Tracking Power (simulation)	10 μW

References

[1] D & R International, Ltd., "2011 Buildings Energy Data Book," for the Buildings Technologies Program Energy Efficiency and Renewable Energy U.S. Department of Energy, 2012

[2] D. Dietrich, D. Bruckner, G. Zucker, P. Palensky, "Communication and Computation in Buildings: A Short Introduction and Overview," *IEEE Tran. on Industrial Electronics*, vol.57, no.11, pp.3577-3584, 2010

[3] Y. M. Chi et al, "CMOS Camera With In-Pixel Temporal Change Detection and ADC," *IEEE JSSC*, vol.42, no.10, pp.2187-2196, 2007

[4] A. G. Andreou, K. Strohbehn, R. E. Jenkins, "Silicon retina for motion computation," *IEEE ISCAS*, pp.1373-1376 vol.3, 1991

[5] R. Etienne-Cummings, J. Van der Spiegel, P. Mueller, "A focal plane visual motion measurement sensor," *IEEE Tran. on Circuits and Systems I: Fundamental Theory and Applications*,vol.44,pp.55-66,1997

[6] H. Jiang, C. Wu, "A 2-D velocity- and direction-selective sensor with BJT-based silicon retina and temporal zero-crossing detector," *IEEE JSSC*, vol.34, no.2, pp.241-247, 1999

[7] B. Zhao, X. Zhang, S. Chen, "A CMOS Image Sensor with on-chip Motion Detection and Object Localization," *IEEE Tran. on Circuits and Systems for Video Technology*, vol. 22, no.4, pp. 581-588, 2012

[8] S. K. Mendis et al, "CMOS active pixel image sensors for highly integrated imaging systems," *IEEE JSSC*,vol.32,no.2,pp.187-197,1997

[9] S. Mitra, G. Indiveri; "A low-power dual-threshold comparator for neuromorphic systems," Research in Microelectronics and Electronics, 2005 PhD vol.2, pp. 198- 201, 2005