A Low-complexity Image Compression Algorithm for Address-Event Representation (AER) PWM Image Sensors

Denis Guangyin Chen, Amine Bermak, and Chi Ying Tsui
The Hong Kong University of Science and Technology, ECE, Hong Kong
Email: {denischn, eebermak, eetsui}@ust.hk

***Abstract*—In Pulse-Width Modulation (PWM) image sensors the incident light intensity is represented by the timing of pulses. Exceptionally high dynamic range (DR) and improved signal-to-noise-ratio (SNR) have been demonstrated for this class of image sensors. Unfortunately, their spatial resolution is limited by the need of an in-pixel memory to record the timing information. The AER protocol is an attractive method for removing this overhead, since pixel trigger events can be sent as address vectors, and in-pixel data memories are no longer required. Regrettably, the need to send address vectors can place an increased burden on the communication channel and will limit the array resolution, frame-rate, and image quality. In this paper, we present a low-complexity AER Block Compression (AERBC) algorithm which exploits the statistically ordered nature of AER pixel arrays. The address vector overhead can be dramatically reduced under this scheme. Only 0.0625 comparisons and 0.125 subtractions are performed for each pixel, and on average 30.82 dB PSNR can be achieved at 1.0 bit-per-pixel code rate. A general strategy is also developed here to optimize AERBC parameters so a balance between performance and hardware resources can be reached.**

I. Introduction

Address-Event Representation (AER) is a communication scheme, where the source address is sent as events trigger [1]. It can be used in Pulse Modulation (PM) image sensors to communicate pixel activity to external circuits outside of the sensor array. In pulse-width modulation (PWM) image sensors, the incident light intensity is encoded by the timing of pulses. The basic principle of this operation is illustrated in Fig.1. To initiate a photo-measurement cycle, like in standard voltage-readout APS pixels, the photo-diode voltage is reset to a defined voltage level V_O by applying a short pulse signal V_{Reset} to a reset switch (transistor). Subsequently, the photo-diode is discharged by the photo-generated current. A comparator continuously compares the integration voltage ramp (V_{int}) to a, usually fixed, voltage reference V_{ref}. When the threshold is reached, a digital pulse (V_{out}) is generated. The integration time is inversely proportional to the photogenerated current - brighter pixels yield shorter integration times and vice versa. Exceptionally high dynamic range (DR) and improved signal-to-noise-ratio (SNR) are immediate benefits of this approach [2].

The main disadvantage of PM image sensors is their lower spatial resolution when compared to APS style CMOS imagers due to the additional complexity in their pixel circuit. Their pixel pitch can be 45 μm in a 0.35 μm technology [2] compared to 7 μm for the classic 3T APS pixel at the same technology node [3]. Some of this circuitry overhead can be mitigated if the memory module is omitted from the pixel. Instead, an in-pixel handshake circuit can be used to communicate edge trigger events to external circuits via the AER protocol (Fig.2). Pixel pitch of 17 μm in a 0.35 μm process has been demonstrated for this class of PM image sensors [4].

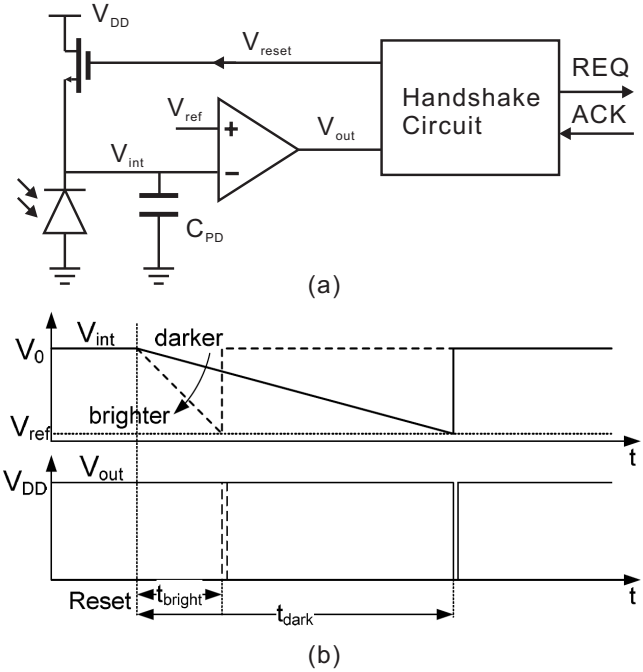

Fig. 1. Principle of PWM time-domain or pulse-modulation (PM) imaging. (a) Pixel circuit architecture an Address-Event Representation (AER) sensor and (b) its operation waveform.

Unfortunately, the adoption of AER protocol means that pixel address bits have to be sent for each pixel. This overhead might be negligible for low-resolution sensors, but for high-resolution sensors, the address vector will easily overshadow the conventional number of bits needed to encode raw light intensity [5]. The storage and transmission of these address bits will become a formidable challenge as the pixel array scales -imposing a limit on the attainable resolution, frame-rate, and image quality.

978-1-4244-9474-3/11/$26.00 ©2011 IEEE

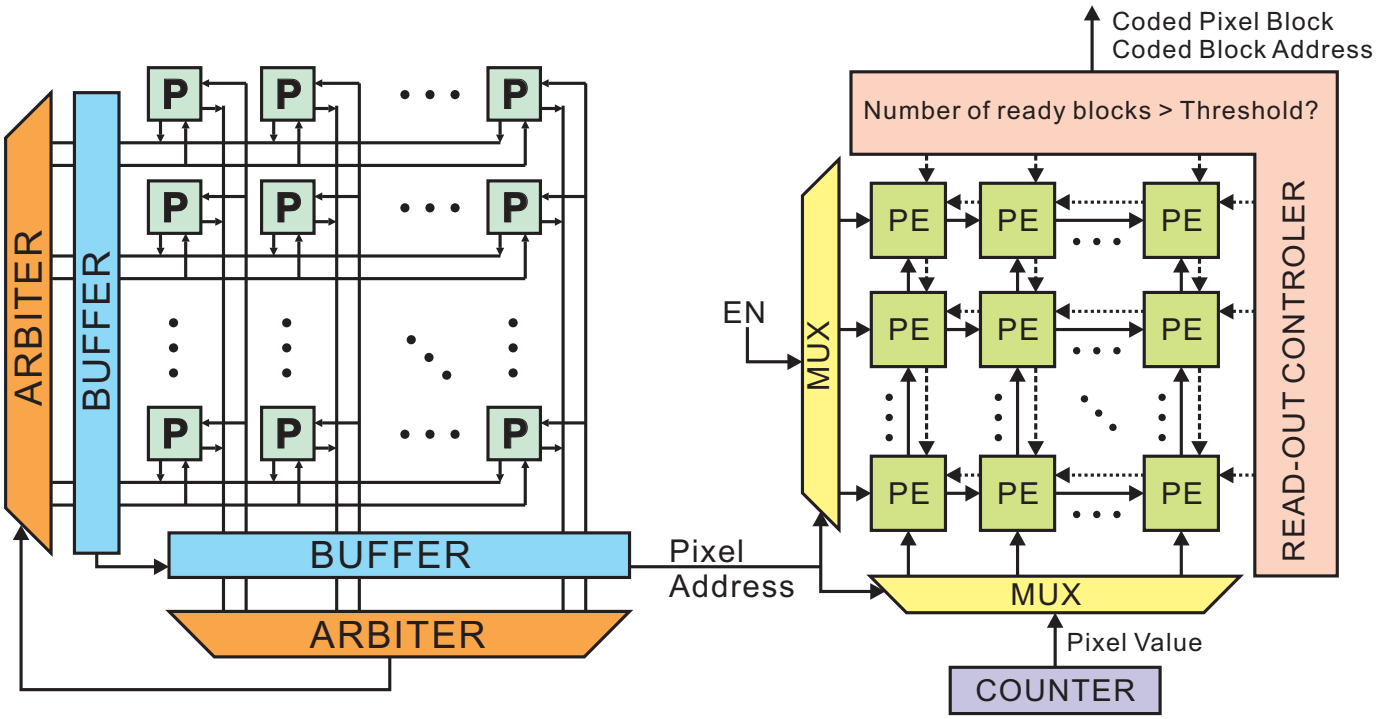

Fig. 2. System level architecture of the proposed compression algorithm.

In this paper, we present a block based image compression algorithm, which exploits the intensity-ordered nature of AER image sensors to simultaneously achieve lower computational complexity and lower address overhead for individual pixels. The main algorithm will be presented in Section II. A general optimization strategy with hardware considerations will be discussed in Section III. Section IV explains the simulation results. Final conclusions will be made in Section V.

II. AER Block Compression

The basic principle of AER Block Compression (AERBC) is to first perform intra-block compression within non-overlapping 4×4 Coded Pixel Blocks (CPB) before coding the address of the CPBs by run-length coding. The system level architecture is shown in Fig.2.

Visual Pattern Image Coding (VPIC) [6] and Block Truncation Coding (BTC) [7] are two classes of image compression algorithms closely related to AERBC. VPIC can achieve high compression ratios (on the order of 0.5 bpp) by classifying non-overlapping 4×4 image blocks into edge or flat blocks. It only transmits the block mean, edge gradient, and bit-pattern index. In the simplest implementation, 2.25 additions, 0.53 comparisons, and 0.1875 multiplications are required to code each image pixel. BTC can typically achieve higher PSNR at the cost of lower compression ratio (on the order of 1.5 bpp) and significantly higher computational complexity. BTC requires the calculation of both the mean and standard deviation within a pixel-block. AERBC can be demonstrated to achieve both high PSNR (comparable to BTC schemes) and high compression ratio (on the order of 1.0 bpp) at much lower hardware complexity than both algorithms. On average only 0.0625 comparisons and 0.125 subtractions are performed for each pixel. This lends AERBC as a promising candidate for ultra-low-energy image compression applications.

A. Intra-block Pixel Coding

The first-order behaviour of PWM AER image sensors can be modelled by

$$T = \frac{(V_{reset} - V_{ref}) \times C_{PD}}{I_{ph}} \tag{1}$$

where T is the time between pixel reset and pixel event trigger, V_{reset} is the pixel reset voltage, V_{ref} is the global reference voltage, C_{PD} is the photo-diode node capacitance, and I_{ph} is the sensed photo-current. It can be deduced here that the pixels will trigger in the descending order of its sensed light intensity.

The coding layers of AERBC is illustrated in Fig.3. Each $N_X \times N_Y$ pixel image is first partitioned into non-overlapping 4×4 pixel blocks called Coded Pixel Blocks (CPB). A group of $N_{MB} \times N_{MB}$ CPBs make up one Macro Block (MB) and a group of $N_{GOB} \times N_{GOB}$ MBs make up one Group of Blocks (GOB). The following parameters are calculated within each CPB (16 pixels):

$$\mu_{med} = \frac{1}{2}(P(8) - P(9)) \tag{2}$$

$$G = \frac{1}{2}(P(1 + SD) - P(16 - SD)) \tag{3}$$

where $P(i)$ is the i-th pixel received in the block, and SD is a tuning variable. The pseudo-gradient, G, is used as a decision

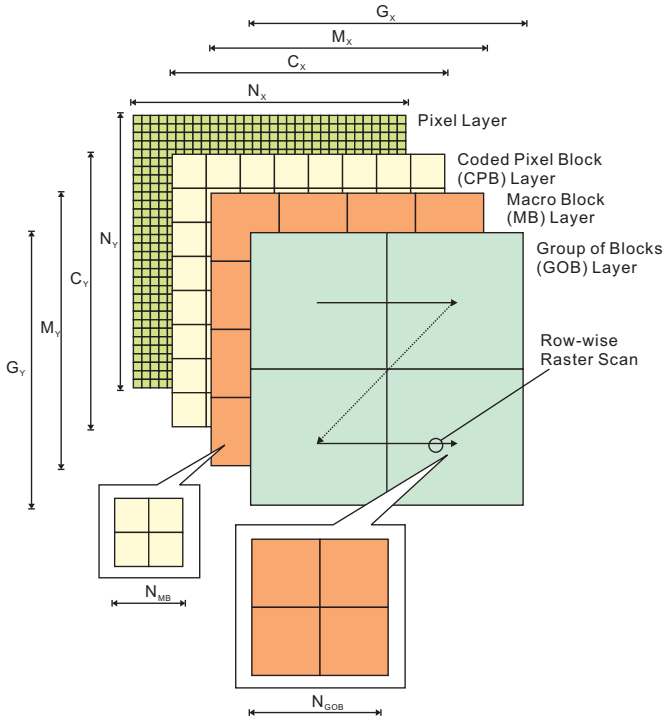

Fig. 3. Coding layer hierarchy from Group of Blocks (GOB) to Coded Pixel Block (CPB).

criteria to distinguish between Edge Patterns (EP) and Uniform Patterns (UP). If G is higher than $threshold$, the block is EP and 1 bit for each pixel is sent along with μ_{med} (quantized to n_{ue} bits) and G (quantized to n_g); otherwise, the block is UP and only μ_{med} (quantized to n_{uu}) is sent. A flag bit is used to indicated wither a block is EP or UP. This algorithm is summarized as followings:

$B(x,y) = -1$, $x \in \{1,2,3,4\}$, $y \in \{1,2,3,4\}$
CPB_READY = FALSE
for $i = 1$ **to** $16 - SD$ **do**
  if $i <= 8$ **then**
    $B(Addr(P(i))) = 1$
  end if
  if i == 9 **then**
    $\mu_{med} = \frac{1}{2}(P(8) - P(9))$
  end if
  if $i == 16 - SD$ **then**
    $G = \frac{1}{2}(P(1+SD) - P(16-SD))$
    CPB_READY = TRUE
  end if
end for
if $G < threshold$ **then**
  CPB_FL = 1, send: μ_{med} in n_{uu} bit resolution.
else
  CPB_FL = 0, send: μ_{med} in n_{ue} bit resolution, G in n_g bit resolution, and B in 4×4 bits.
end if

The $Addr(P)$ operator returns the spatial address of pixel P, which is the output of the AER arbitration process. After optimizing over a set of standard test images, $SD = 4$, $threshold = 12$ (for intensity values ranging from 0 to 255), $n_{uu} = 7$, $n_{ue} = 4$, and $n_g = 4$, is found to yield good results.

It can be observed that only 1 comparison and 2 subtractions are performed for each pixel block, and 32 bits of memory is needed in each PE to store the intermediate results. This leads to smaller hardware footprint, and potentially, higher speed and lower energy consumption.

Once the compressed image data is collected at the receiver's end, the raw image can be assembled as either a uniform block with value μ_{med} or the sum of a uniform block with value μ_{med} and the bipolar bit-pattern, B, scaled by the pseudo-gradient coefficient, G. This is summarized as followings:

if CPB_FL == 1 **then**
  $I(x,y) = \mu_{med}$, $x \in \{1,2,3,4\}$, $y \in \{1,2,3,4\}$
else
  $I(x,y) = \mu_{med} + B(x,y) \times G$, $x \in \{1,2,3,4\}$, $y \in \{1,2,3,4\}$
end if

for every CPB, where I is the 4×4 pixel value matrix, and the final decoded image is obtained by performing a simple 3-by-3 averaging filter over the raw image.

B. Inter-block Address Coding

Redundancy in the address vector can be suppressed if several CPBs are sent together. Their relative addresses can be coded by run-length coding. This concept is illustrated in Fig.4, where row-wise raster scan is used (Fig.3). As pixels are received from the AER pixel array, more and more CPBs become ready to be sent ($16 - SD$ pixels are needed in each 4×4 pixel CPB before the block can be coded).

When $TXthreshold$ number of CPBs are ready (sum of (CPB_READY == TRUE)), the read-out controller in Fig.2 initiates a raster scan across the processing elements. If no ready CPBs are found in any consecutive GOBs, the *GOB_FL* is set to 1, and *GOB_RUN* encodes the number of CPB-free GOBs. If the run count exceeds the maximum allowable counter value ($2^{n_{RL}}$) in *GOB_RUN*, it is encoded over several consecutive *GOB_FL* and *GOB_RUN* packets.

If ready CPBs are found in a GOB, *GOB_FL* is set to 0, and the MBs under that GOB is searched using the same run-length coding scheme as at the GOB level. When a ready CPB

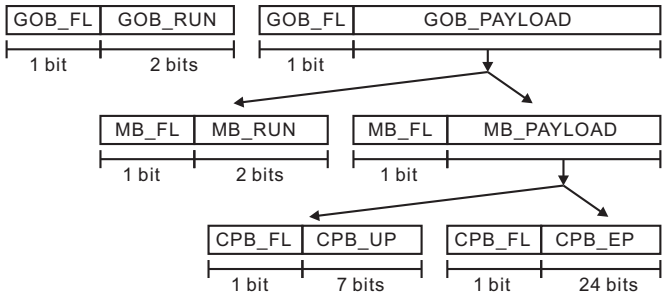

Fig. 4. Coding protocol of Group of Blocks (GOB) and Coded Pixel Block (CPB).

Fig. 5. AERBC result for the 512×512 grey-scale test image of Lena with 0.9 bits-per-pixel and 30.66 dB PSNR.

is encountered in a MB scan, *MB_FL* is set to 0 and the CPB is coded using the algorithm described in Section II-A.

III. Hardware Resource Considerations

The value of $TXthreshold$, the number of MB in each GOB (N_{MB}), and the run-length counter bit-width (n_{RL}) can be optimized for maximum compression ratio. Setting a higher $TXthreshold$ will typically yield a higher compression ratio, but this has to be considered against limitations in hardware resources. Therefore the buffer may be full at some point and no more CPBs can be buffered, and the request queue from the AER pixel array will be stalled.

Any buffered CPBs in the PE array, which are ready to be sent will have to be sent at this point, even though it may cost additional address vector overhead. If further room is needed in the PE array, CPBs will have to be encoded and sent away prematurely before $16 - SD$ pixels have been received. This will come at a cost of lower PSNR at the receiver's end. An important question to ask at this point is how many PEs are needed to prevent this scenario from occurring? Here, we assume that an infinite number of PEs are available, and the control unit is ideal where it is able to utilize all PEs. In practice, PE utilization will never reach 100% since utilization efficiency must be traded for PE allocation speed. This trade-off will improve as more efficient PE allocation algorithms are developed.

In simulations, it is possible to monitor the number of PEs occupied during the frame read-out. The tuning parameters can be optimized to seek both lower peak PE occupancy and higher compression ratio. Using simulations across a set of standard test images, $TXthreshold = 3000$, $N_{MB} = 4$, and $n_{RL} = 2$ is found to yield reasonable results across the spectrum.

IV. Simulation Results

The AERBC algorithm described in Section II is applied to a database of 512×512 test images (Table I). The PSNR of the decoded image, peak number of PEs occupied during one read-out frame, bits-per-pixel (bpp), and compression ratio ($CR = (9 + 9)/bpp$) are reported in Table I.

It can be observed that AERBC performance improves for images with less high frequency content at its global level.

Image	PSNR (dB)	Peak PE Occupancy	bpp	CR
Cameraman	30.39	6591	0.9	20.0
Lift-body	33.24	8226	0.7	25.7
Lena	30.66	6996	0.9	20.0
Mandrill	26.00	11210	1.4	12.9
Pirate	28.20	8331	1.1	16.4
Woman-dark-hair	36.44	5953	0.8	22.5
Average	30.82	7885	1.0	19.6

TABLE I
Summary of AERBC simulation results on standard test images. Bits-per-pixel (bpp) includes both data and address bits.

The implications here are two folds: for images with more flat areas, the errors resulting from intra-block compression is lower, and on top of that pixels in close spatial proximity will trigger at close temporal proximity which leads to lower peak PE occupancy because PE buffers are purged more frequently.

V. Conclusion

The AER protocol is an attractive method for minimizing pixel circuits in PWM image sensors to achieve higher spatial resolution. Unfortunately, this brings about a significant address vector overhead which scales with sensor resolution. AERBC is developed to alleviate this overhead. It exploits the statistically ordered nature of AER pixel arrays to achieve very low computational complexity, and the simulation results verify that good PSNR figures can be obtained at high compression ratio. A general strategy has also been outlined to optimize AERBC parameters so a balance between performance and hardware resources can be reached.

Acknowledgement

The authors would like to thank the Hong Kong Research Grant Council for their support on this work under grant reference 610509.

References

[1] K. Boahen, "A burst-mode word-serial address-event link-i: transmitter design," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 51, no. 7, pp. 1269–1280, July 2004.

[2] A. Kitchen, A. Bermak, and A. Bouzerdoum, "Pwm digital pixel sensor based on asynchronous self-resetting scheme," *Electron Device Letters, IEEE*, vol. 25, no. 7, pp. 471–473, July 2004.

[3] S. Hamami, L. Fleshel, and O. Yadid-Pecht, "Cmos aps imager employing 3.3 v 12 bit 6.3 ms/s pipelined adc," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 4, 23-26 2004, pp. IV – 960–3 Vol.4.

[4] C. Shoushun and A. Bermak, "Arbitrated time-to-first spike cmos image sensor with on-chip histogram equalization," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 15, no. 3, pp. 346–357, March 2007.

[5] E. Culurciello, J. H. Park, and A. Savvides, "Address-event video streaming over wireless sensor networks," in *Circuits and Systems, 2007. ISCAS 2007. IEEE International Symposium on*, May 2007, pp. 849–852.

[6] D. Chen and A. Bovik, "Visual pattern image coding," *Communications, IEEE Transactions on*, vol. 38, no. 12, pp. 2137 –2146, dec. 1990.

[7] E. Delp and O. Mitchell, "Image compression using block truncation coding," *Communications, IEEE Transactions on*, vol. 27, no. 9, pp. 1335 – 1342, sep. 1979.