FRAME BASED ADAPTATION FOR CMOS IMAGER

Dave Climie, Shoushun Chen, Amine Bermak

Hong Kong UST, EEE Department,
Clear water Bay, Hong Kong
dclimie@ust.hk, dazui@ust.hk, eebermak@ust.hk

Dominique Martinez

LORIA-INRIA, Nancy, France
dominique.martinez@loria.fr

Abstract—this paper presents a VLSI architecture of a frame-based adaptive quantization technique based on the Fast Boundary Adaptation Rule (FBAR). The adaptive quantization algorithm is integrated together with a 128x128 pixel CMOS image sensor array. The pixel operation is based on biologically inspired time-to-first spike encoding scheme. System and circuit level simulations show the successful operation of the proposed architecture.

Keywords—Adaptive Quantization, PWM CMOS Image Sensor

I. INTRODUCTION

Digital image sensors continue to grow in popularity, with CMOS photo-sensors increasing in importance due in large part to the advances of CMOS process technology. The CMOS photo-sensor in its most common configuration acts as a current source with output proportional to incident illumination. A variety of mechanisms have been devised to convert this analog representation of intensity into a digital value. Standard A/D conversion of CMOS sensor pixel intensity information is performed by a regular N-point scalar quantizer. The input range of the analog signal is divided into N contiguous and linearly spaced intervals, or bins, allowing for conversion of the analog value to a digital representation among a finite set of N discrete symbols. An improvement to the standard fixed scalar quantizer has been explored in various domains and is termed 'adaptive quantization'. A given quantizer is optimal if it minimizes some measure of distortion between the original analog signal and the corresponding set of digital codes. The potential benefits of adaptive quantization applied to image quantization are substantial and include increased resolution for a fixed-length binary code, or improved compression of digital codes for a fixed resolution requirement.

This paper presents a frame-based image sensor implementation of a previously established unsupervised competitive learning rule termed FBAR, or Fast Boundary Adaptation Rule [1], [2]. Variants of FBAR have been shown to obtain near optimal performance with respect to minimization of mean absolute error (MAE) and mean squared error (MSE) quantization distortion. In this study, a frame-based version of FBAR has been integrated into a time-based A/D converter and coupled directly to an on-chip 128x128 pixel CMOS image sensor array.

A description of the image sensor with integrated FBAR is given in Section II. Section III introduces the FBAR algorithm and discusses hardware implications. Simulation results are provided in Section IV, with a conclusion in Section V.

II. SYSTEM DESCRIPTION

A block diagram of the image sensor IC with integrated adaptive quantizer is shown next:
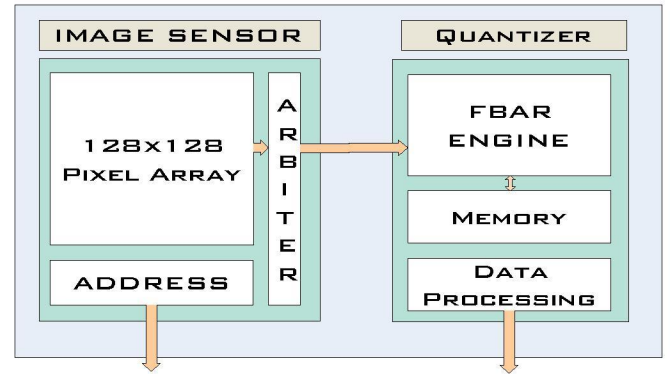

Figure 1 - CMOS Image Array + Quantizer

The image sensor implementation is based on an analog pulse width modulated (PWM) architecture, which provides several compelling advantages over standard active pixel sensor (APS) or pulse frequency modulating (PFM) techniques [3]. The operation of the PWM sensor is described as follows: A global reset pre-charges the full array of photodiodes to a common pre-defined voltage level V_{RESET}. The release of the reset signal begins an integration phase, with light exposure resulting in a photo-current I_D and thus discharging each internal photodiode capacitance C_D at a rate dependent on individual pixel input illumination. After some amount of time each node crosses a fixed threshold voltage V_{TH}. The resultant integration time is then digitized, and intensity information can be derived.

When operated in conjunction with a fixed scalar A/D quantizer, data intensity assignment is performed in a fixed fashion, with an established set of digital codes assigned to each particular contiguous region of pixel expiry times. The non-linear relationship between the expiry times of each pixel

requires compensation, which is readily performed by adjusting time bins at a variable rate dependent on the position within a frame. The addition of adaptive quantization will demand the adjustment of these time/intensity bins based on the statistics of the input image. Pixel voltage discharge and data quantization is demonstrated in **Figure 2**.
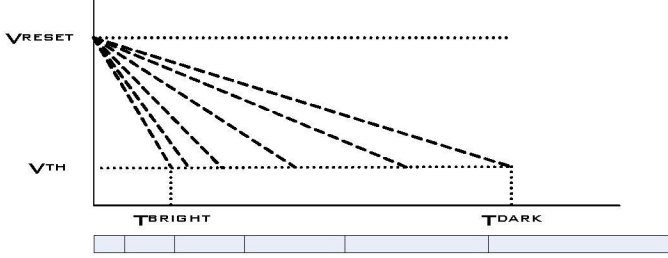

Figure 2 - PWM APS Sensor Operation

III. FBAR IMPLEMENTATION

A scalar quantizer is tasked with the translation of an analog input signal x into a discrete set of values y_i, $i = 1, \ldots, N$. An optimal quantizer attempts to minimize a given measure of distortion $d(x, Q(x))$ of the resulting codes. The FBAR algorithm [1],[2] is based on minimizing the Holder norm and the rth power law distortion:

$$d(x, Q(x)) = Dr = \sum_{i=1}^{N} \int_{x_{i-1}}^{x_i} |x - y_i|^r p(x) dx \qquad (1)$$

The input analog range is divided into N contiguous and exhaustive intervals $R_i \equiv [x_{i-1}, x_i)$ each of with boundary points:

$$x_0 = -L < \ldots < x_{i-1} < \ldots < x_N = L$$

The FBAR algorithm is performed in the following manner: at each input event x falling within R_k, each R_i $i = 1, \ldots, N$ is modified by adjusting each boundary point by Δx_j according to:

$$\Delta x_j = \eta \left(\sum_{k=j+1}^{N} \delta_k^r \frac{\|_{Rk}}{N-j} - \sum_{k=1}^{j} \delta_k^r \frac{\|_{Rk}}{j} \right) j = 1, \ldots, N-1 \quad (2)$$

Where $\|_{Ri} = 1$ for $x \in R_i$, and 0 otherwise, $\delta_i = x_i - x_{i-1}$, and η is the learning rate.

The implications of (2) with regards to hardware implementation complexity are two-fold. First of all, up to N memory accesses will be required for each boundary adjust operation in order to access the stored values of δ_j^r. In

addition, it is apparent that each boundary adjustment Δx_j for $j = 1, \ldots, N-1$ will require up to N multiplications and $N-1$ additions of each regional hit count, for a maximum total of $\approx 2N^2$ operations per image frame. Although this processing requirement could be satisfied for moderate sized N by adding parallel processing elements or a fast system clock rate, power consumption and/or processing speed would grow impractical for a large N, a problem that would be especially pronounced for high frame rates.

We can simplify (2) by setting $r = 0$ to obtain:

$$\Delta x_j = \eta \left(\sum_{k=j+1}^{N} \frac{\|_{Rk}}{N-j} - \sum_{k=1}^{j} \frac{\|_{Rk}}{j} \right) j = 1, \ldots, N-1 \qquad (3)$$

It has been shown in [2] that Eq. (3) leads to an equi-probable quantization, i.e. p(Ri)=1/N for all i. We rearrange to obtain a result that is readily implemented in hardware, with computation cost reduced from $O(N^2) \Rightarrow O(N)$:

$$\Delta x_j = \eta \left(\frac{\sum_{k=j+1}^{N} \|_{Rk}}{N-j} - \frac{\sum_{k=1}^{j} \|_{Rk}}{j} \right) j = 1, \ldots, N-1 \qquad (4)$$

A. Frame-Based Operation

Prior applications of the FBAR algorithm have employed a per-event update mechanism [1], termed 'on-the-fly', with each boundary point adjusted after the reception of each individual input event. Lending itself to a simple hardware implementation, we propose a per-frame update strategy. Boundary intervals will remain static on a per-frame basis, with input image statistics affecting interval boundary points only for the *next* frame. In this manner, memory updates are reduced from a pixel-based implementation, and memory accesses occur in a natural fashion with increasing time (decreasing intensity) without the need to adjust previously assigned intensity data values to expired events.

Let the set of intensity bins at time t be represented as $R_k(t)$. We calculate the set of updated values of $R_k(t+1)$ as follows:

$$R_k(t+1) = (R_k(t) + \Delta x_k(t) - \Delta x_{k-1}(t)) \quad k = 1, \ldots, N \quad (5)$$

For $j = 1, \ldots, N-1$, $\Delta x_j(t)$ is calculated according to (4), and otherwise $\Delta x_0(t) = \Delta x_N(t) = 0$.

B. System Architecture

The proposed FBAR system was implemented in Verilog HDL and coupled with a custom CMOS 128x128 pixel image sensor. This section presents a general outline of the FBAR system architecture.

The front end processing block accepts pixel input event pulses from the 128x128 image sensor array. The front end is tasked with generating a binary count value tracking the number of incident events since the initialization of the frame to be utilized by the subsequent FBAR engine. Input pixel event pulses feed from the front end image sensor array at a rate of $128 \times 128 = 2^7 \approx 16,000$ events per frame. In order to handle the asynchronous nature of the input hit pulse with respect to the system clock, a gray code is used in the front end counter, which is then fed into a pipeline register which performs clock synchronization. A gray to binary converter is placed after the input synchronization register to provide binary inputs to further processing stages.

The fundamental data type of an image quantizer is the set of disjoint and exhaustive intensity quantization intervals, or bins. Rather than employing a static set of quantization bins as is the case for a linear quantizer, the FBAR quantization process is tasked with the continuous update of the boundary points of these bins.

It is proposed that a local memory act as a central repository of intensity quantization interval data, and that each memory location store an intensity magnitude. This concept along with an example adaptation cycle is shown in **Figure 3**.
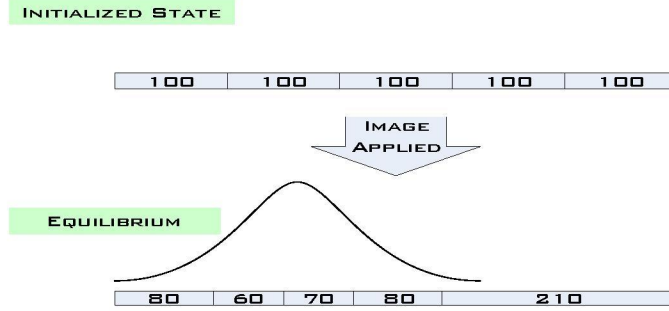

Figure 3 - Intensity Bins Adapting to Input Image

A central controller is tasked with overall system control and coordination as well as with general time keeping. The front-end processing block is initiated and tracks a binary count of incoming pulses for the duration of each frame. At each bin interval expiry, the count value is latched for input to the FBAR engine. The controller contains a general system timer tracks the current δ_i, or bin interval, which before the beginning of each interval is pre-fetched from memory via the memory controller.

Upon completion of a bin, the central controller then initiates the FBAR engine to calculate Δx_j according to Equation (4). The central controller performs an interval update (5) using two previously computed Δx_j values from the current and previous bin time interval, and upon completion stores the result back into SRAM for use with the next input frame of data.

This process repeats for each bin, and begins with each frame anew following the completion of a frame. Bin interval

δ_i adjustment will therefore continue indefinitely, either reaching equilibrium with a fixed input frame, or a series of input images can be fed into the system, which will cause continually adapting intensity intervals.

C. Macro-Bins

In an effort to reduce the computational requirements per image frame, the concept of a macro-bin was introduced. A macro-bin is defined as a superset interval consisting of some number P>1 of fixed and linearly spaced micro-bins. In the case of a 256-level intensity scale divided by P, the FBAR algorithm is tasked only with macro-bin boundary adjustment, in effect reducing the number of boundary points by a factor of P. As each macro-bin is adjusted, all P micro-bins within it are adjusted by the same factor, with each micro-bin thus maintaining a size of $1/P^{th}$ of their corresponding macro-bin. A value of P=4 was selected for implementation into the target IC, thereby resulting in a total of 64 macro-bins for FBAR adjustment, each containing 4 linearly spaced micro-bins.

D. Support Circuitry

An SRAM block of size 128x8 bits is employed to provide for 64 macro-bin intervals δ_i, each of 16-bits wide. A memory controller multiplexes access from the internal central controller and/or an external chip interface. The direct-SRAM interface allows SRAM values of δ_i to be initialized to a reset state by an external interface prior to execution.

A register bank stores configuration data such as η and maximum overflow / underflow targets that will be used to control the rollover of FBAR calculations.

An 8x8 MUX can be externally controlled in order to monitor from a selection of internal signal lines. Quantization count status, intermediate FBAR calculation results, and SRAM control signals can each be selected for observation from an external 8-bit bus.

IV. SIMULATION RESULTS

System-level Matlab and HDL simulations were performed to verify system performance and evaluate alternative architectures.

A. System Level Simulations

The image sensor and FBAR system were modeled in Matlab prior to hardware implementation in order to evaluate important performance metrics such as convergence speed.

The following graphic presents an example simulation demonstrating the convergence of image intensity bins to a state of equilibrium:
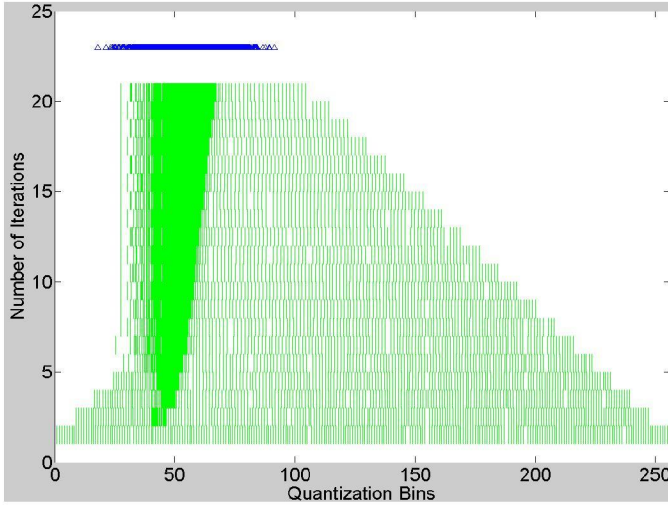
Figure 4 – Quantization Bin Convergence

Image intensity is represented on the x-axis, which is divided into 256 intensity bins, which are equally spaced at system initialization (y=0). 20 image frames are processed, with boundary points updated after each frame. In this example, an input image is generated in the lower portion of the intensity scale (20-100). Boundary points are seen to converge about the image with increasing frame iterations as expected.

The speed of system equilibrium is found to be highly dependent on the selected learning rate η The convergence of quantizer boundary points towards the highest density of image events is increased in speed with a higher η, however this results in an increase in jitter at equilibrium. Hardware simulations will be performed to optimize η for a given input image, and adapting η according to the dynamics of the input scene will be further evaluated.

B.   HDL Simulation

Full system Verilog HDL simulations were performed in order to prove a functioning FBAR system prior to the release of silicon. A Verilog testbench integrates a comprehensive memory initialization and verification routine, with input image sequences and corresponding expected intensity interval values.

A sample simulation of a 4-frame FBAR simulation is shown next. An input image with a high density of intensity events centered in the brightest half of the frame is input to the FBAR HDL model. Frames are shown increasing in time along the x-axis. The top simulation line shows the high density input intensity events centered about the bright portion of an image frame. The bottom line shows the 64 macro-bin intensity intervals, initialized as equidistant intervals, which then converge to the higher density region of pixel events after subsequent frame iterations.
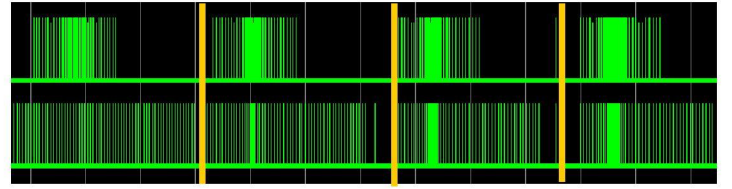

Figure 5 - Verilog Simulation – 4 Frames

A zoomed-in view of the increased density of both macro and micro-bins about a high-density region of pixel events is shown in the following figure:
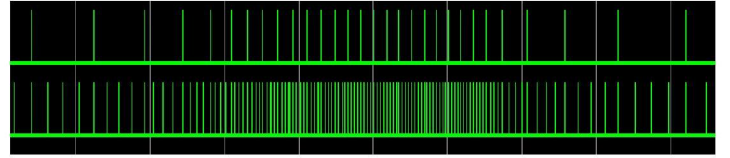

Figure 6 - Verilog Simulation - 1 Frame

Each macro-bin can be seen to contain a set of 4 micro-bins that remain equal in size with the expansion of their parent macro-bin.

V.   CONCLUSION

An image sensor with an integrated quantizer based on the FBAR algorithm has been presented. Pixel intensity intervals are represented in the time domain. Adaptation of quantization time intervals is performed according to the FBAR algorithm, with continuous adjustment of intensity intervals updated in a per-frame manner. Matlab and Verilog simulations have proven that the system functions as intended, and can acquire a reasonable convergence speed with an appropriate learning rate. Hardware silicon results will be presented in the future.

REFERENCES

[1]   Van Hulle, M.M., "Fast Adaptive Scalar Quantization for Minimal Mean Squared Error Distortion in the High Resolution Case: Extension of the Boundary Adaptation Rule," *IEEE World Congress on Computational Intelligence, Orlando, FA*, 4365-4369, 1994.

[2]   D. Martinez and M. Van Hulle, "Generalized Boundary Adaptation Rule for Minimizing rth Power Law Distortion in High Resolution Quantization," *Neural Networks, vol. 8, No. 6, 1995*

[3]   A. Kitchen, A. Bermak and A. Bouzerdoum, "PWM DPS Based on Asynchronous Self-Resetting Scheme,' *IEEE Electron Device Letter, vol. 25, No. 7, 2004*

[4]   C. Shoushun and A. Bermak, "A low power CMOS imager based on time-to-first-spike encoding and Fair arbitration," ISCAS2005, pp. 5306-5309, Kobe, Japan, 2005.