A CMOS Image Sensor with on Chip Image Compression based on Predictive Boundary Adaptation and QTD Algorithm

Chen Shoushun†, Amine Bermak† , Wang Yan† and Dominique Martinez‡
†Smart Sensory Integrated Systems Lab, EEE Department
Hong Kong University of Science and Technology, Hong Kong, SAR.
‡LORIA-CNRS, Vandoeuvre-Les-Nancy 54506, France.

***Abstract*— This paper presents the architecture, algorithm and VLSI hardware of image acquisition, storage and compression on a single-chip CMOS image sensor. The image array is based on time domain digital pixel sensor technology equipped with non-destructive storage capability using 8-bit Static-RAM device embedded at the pixel level. An adaptive quantization scheme based on Fast Boundary Adaptation Rule (FBAR) and Differential Pulse Code Modulation (DPCM) procedure followed by an on-line Quadrant Tree Decomposition (QTD) processing is proposed enabling low power, robust and compact image compression processor. A prototype chip including 64×64 pixels, read-out and control circuitry as well as the compression processor was implemented in $0.35\mu m$ CMOS technology with a silicon area of $3.2 \times 3.0mm^2$. Simulation results show compression figures corresponding to 0.75 Bit-per-Pixel (BPP), while maintaining reasonable PSNR levels.**

I. Introduction

Real time image acquisition and processing is becoming a challenging task because of higher spatial and coding resolution, which imposes very high bandwidth requirement. The recent emergence of new applications in the area of wireless video sensor network and ultra low power biomedical applications (such as the wireless camera pill) have created new design challenges in which the hardware is often constrained to take very little physical space and to consume very little power. However, image compression remains the most expensive hardware [1][2][3] in digital video camera. This would limit the prospect of implementing low power image acquisition and compression on a single chip. In order to alleviate some of these problems, we reported a single chip vision sensor based on Fast Boundary Adaptation Rule (FBAR) followed by an on-line Quadrant Tree Decomposition (QTD) processing [4] enabling low power and compact image compression. The image is first acquired using a time domain CMOS digital pixel sensor array followed by FBAR scheme which permits to compress the data to 1 Bit-per-Pixel (BPP). Further compression ($0.6 - 0.8$ BPP) is accomplished using QTD algorithm. The scanning sequence in this work is based on a Morton (Z) [5] scan strategy which is a quadrant or window-based read-out featuring extremely compact hardware implementation by bitwise address manipulation. However, the transition from one quadrant to the next involves jumping to a non-neighboring pixel resulting in spatial discontinuity. We proposed a smooth boundary point propagation scheme but at the expense of additional two 8-bit registers for each level quadrant.

In this paper, we propose a second generation of image compression system. Compared to the previous work, the 1-bit FBAR algorithm is performed on the predictive error using DPCM rather than the pixel itself. A new Hilbert scanning technique [6] is used avoiding any spatial discontinuity in the scanning sequence and maintaining block based scanning strategy. This makes it quite suitable to both the FBAR and QTD algorithm. Another significant improvement is related to the QTD algorithm which saves silicon area by storing the tree information in the DPS array. The remainder of the paper is organized as follows. Section II introduces design of the TFS-based DPS pixel. Section III introduces the algorithmic considerations for the 1-bit FBAR algorithm combined with the DPCM technique and the simulation results. Section IV describes the imager architecture and its VLSI implementation. Section V concludes this work.

II. Time-to-first spike digital pixel

The image array consists of 64×64 digital pixel sensors. As shown by Fig. 1, each pixel includes a photodiode *PD* with its its internal capacitance C_d, a reset transistor *M1*, a comparator *(M2-M6)* and 8-bit SRAM, each of which is implemented by 9 transistors.

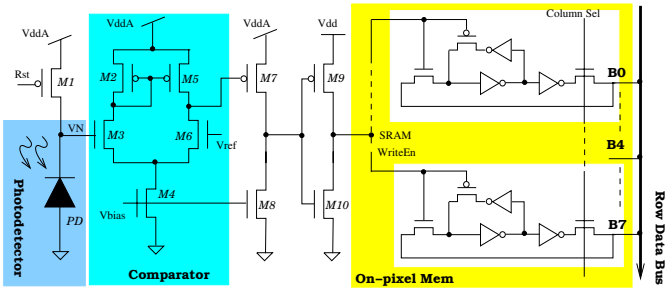


Fig. 1. Pixel Schematic.

The pixel array is operated in two separate phases. The first phase corresponds to the integration phase in which the photodiode is first reset to $VddA$. After that, the light falling onto the photodiode discharges C_d, resulting in a decreasing voltage VN across the photodiode node. The time required

for VN to reach the threshold voltage V_{ref} of the comparator can be interpreted as the time-to-first spike, which is inversely proportional to the photocurrent (I_{ph}), given the assumption of constant I_{ph} and C_d. At this time, the "SRAM WriteEn" becomes invalid, therefore a time stamp provided by a global timing unit (gray de-counter) is recorded into the on-pixel SRAM. Actually, the output of the comparator serves as a "Write-Stop" signal which prevents the SRAM from writing new timing data and hence the last data recorded in the SRAM is the timing stamp for that pixel. Once the integration phase is completed, the pixel array can be interpreted as a distributed static memory. Image processing can be performed by scanning the array using row and column addressing technique.

III. Algorithmic Considerations

A. Adaptive quantization based on FBAR

The proposed adaptive quantizer can be specified by an ordered set of boundary points $y_0 < y_1 < \cdots < y_{i-1} < y_i < \cdots < y_{N-1} < y_N$ delimiting N disjoint quantization intervals $R_1, \cdots, R_i, \cdots, R_N$, with $R_i = [y_{i-1}, y_i]$. The quantization process is a mapping from a scalar-valued signal x into one of reconstruction intervals, i.e., if $x \in R_j$, then $Q(x) = y_j$. This Quantization process thus inevitably introduces quantization error when the number of quantization intervals is less than the number of bits needed to represent any element in a whole set of data. The most commonly used distortion measure is the r^{th} power law distortion:

$$d\left(x, Q\left(x\right)\right) D_r \equiv \sum_{i=1}^{N} \left|x - y_i\right|^r p(x) dx \qquad (1)$$

It has been shown that using Fast Boundary Adaptation Rule [7] can minimize the r-th power law distortion, e.g. the mean absolute error when $r = 1$ or the mean square error when $r = 2$. At convergence, all the N quantization intervals R_i will have the same distortion $D_r(i) = D_r/N$. This property guarantees an optimal high resolution quantization. For a 1-bit quantizer, there will be just one adaptive boundary point y delimiting two quantization intervals, with $R0 = [0, y]$ and $R1 = [y, 255]$. The boundary point itself is taken as the reconstructed value. At each time step, the input pixel intensity will fall into either $R0$ or $R1$. The boundary point is then shifted to the direction of the active interval by a quantity η.

The performance of the 1-bit FBAR quantizer is found highly dependent on a particular choice for η [4]. We propose to make η adaptive using the following heuristic rule: if the active quantization interval does not change between two consecutive pixel readings, we consider that the current quantizing parameters are far from the optimum and η is then multiplied by $\Lambda > 1$; if the active quantization interval changes between two consecutive pixel readings, we consider that the current quantizing parameters are near the optimum and thus η is reset to its initial value.

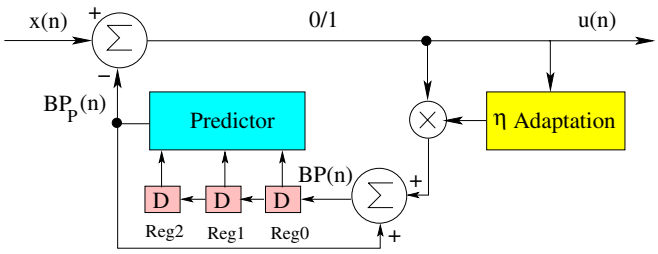


Fig. 2. 1-bit Adaptive Q combined with DPCM.

B. Differential Pulse Code Modulation

In our proposed image compression scheme, the 1-bit adaptive FBAR is incorporated with the DPCM algorithm. The main structure of the hybrid system is shown in Fig. 2. The input signal to our system is the data from an image sensor. Each pixel's value is first compared with its predictive value which is estimated by the equation:

$$BP_P = Reg0 \times 1.375 - Reg1 \times 0.75 + Reg2 \times 0.375 \quad (2)$$

where $Reg0, Reg1, Reg2$ are three previously scanned pixels' reconstructed value. The predictive value is then adjusted by η depending upon the comparison result to obtain its reconstructed value, which is then feed back to the predictor in the next cycle. In this system, both the decoder and the encoder are based on the same mechanism thus no side information is need to be transmitted.

C. Hilbert scan

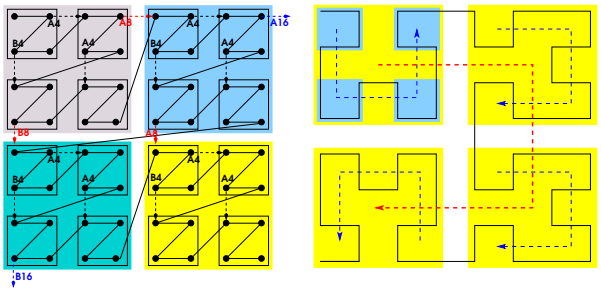
The adaptive quantizer explained earlier permits to build a binary image on which quadrant tree decomposition (QTD) can be further employed to achieve higher compression ratio. The QTD compression algorithm is performed by building a multiple hierarchical layers of a tree which corresponds to a quadrant in the array. In the previous work [4], we built the tree by scanning the array using the Morton (Z) [5] scan strategy as shown by Fig. 3.(A). While it provides a simple mean to scan the array in a block based approach, the transition from one quadrant to the next involves jumping to a non-neighboring pixel which will result in spatial discontinuity, which can be larger and larger when scanning the array due to the inherent hierarchical partition of the QTD algorithm. To address this problem, we proposed a smooth boundary point propagation scheme, at the expense of two additional 8-bit registers for each level of quadrant. As shown in Fig. 3.(A), two registers $(A4, B4)$ are needed to store the boundary point for the 4×4 quadrant level and two other registers $(A8, B8)$ are needed to store those related to the 8×8 quadrant level.

Hilbert scanning provides another interesting solution without using the additional storage. As shown in Fig. 3.(B), the scanning is also performed within multi-layers of quadrants but always keeping spatial continuity when jumping from one quadrant to another. It ensures minimal storage requirement for the adaptive quantizer as the neighboring pixel is the one just scanned. The implementation of Hilbert scanning can be quite straightforward by using hierarchical address mapping logic which will be explained in the next section.

TABLE I

AVERAGE PERFORMANCE OF 20 TEST IMAGES UNDER DIFFERENT OPERATING MODES, NAMELY ADAPTIVE η RASTER SCAN (η-R), ADAPTIVE η MORTON (Z) SCAN (η-MZ), ADAPTIVE η SMOOTH BOUNDARY MORTON (Z) SCAN (η-SMOOTHMZ), ADAPTIVE η HILBERT SCAN (η-HILBERT) AND ADAPTIVE η WITH DPCM USING HILBERT SCAN (η-HILBERT+DPCM). $M = \frac{PSNR}{BPP}[dB/BPP]$. THE TABLE SHOWS THAT EACH MODE CAN ACHIEVE TO ITS MAXIMUM PSNR USING A SPECIFIC VALUE OF η_0. THE η-HILBERT+DPCM MODE PRESENTS THE BEST PSNR AND BPP FIGURES WHEN η_0=18.

Operation modes	Size of test images															
	64×64				128×128				256×256				512×512			
	PSNR	BPP	M	η_0	PSNR	BPP	M	η_0	PSNR	BPP	M	η_0	PSNR	BPP	M	η_0
η-R	21.15	1.07	19.79	21	22.42	0.97	23.05	20	24.01	0.93	25.95	16	25.93	0.91	28.52	12
η-MZ	21.48	0.97	22.15	18	22.82	0.89	25.64	16	24.32	0.84	28.78	13	25.90	0.85	30.55	11
η-SmoothMZ	22.37	0.96	23.37	17	23.83	0.90	26.51	14	25.49	0.86	29.79	12	27.45	0.89	30.76	9
η-Hilbert	22.77	0.98	23.11	18	24.26	0.93	26.05	14	26.05	0.88	29.70	12	28.11	0.99	28.27	7
η-Hilbert+DPCM	**23.06**	0.88	26.14	19	**24.62**	0.81	30.43	15	**26.52**	0.75	35.25	12	**28.53**	0.75	38.02	9



(A.) (B.)

Fig. 3. (A.)Smooth boundary point propagation scheme using Morton (Z) scan. (B.)Hilbert scan patterns at each hierarchy for an 8×8 array.

D. Simulation Results

We have compared the performance of our proposed algorithm with other operation modes over a set of test images. As the performance of FBAR is highly dependent on the choice of η, there exists an optimal η value for a particular image and for each operation modes. For the set of test images, we sweep the value of η from 5 to 35 to find the the optimal value for each operation mode. The performance of each mode is reported in Table I. We can note that using Morton (Z) scan can achieve better performance than raster scan because it is a block based strategy. Further improvement in PSNR of about 1.5dB (512×512) can be achieved by using Smooth Morton (Z) scan. Hilbert scan can easily achieve comparable performance without additional storage requirement. Finally, the best figures of PSNR and BPP are achieved using our proposed hybrid system, combining DPCM with Hilbert scan.

IV. VLSI IMPLEMENTATION

A. Imager Architecture

Fig. 4.(A) shows the block diagram of a single chip CMOS image sensor with the adaptive DPCM quantizer and the QTD processor. The image array consists of 64×64 digital pixel sensors. The pixel array is operated in two separate phases. The first phase corresponds to the integration phase in which the illumination level is recorded and each pixel sets its own integration time which is inversely proportional to the photocurrent. A timing circuit is added in our imager in order to compensate for this non-linearity by adjusting the quantization levels of a sampling counter. In our prototype, a $16bit \times 256word$ memory is used in order to apply more compensation modes than the linear relationship, logarithmic relationship, for instance. In the integration phase, the row buffers drive the timing information in gray code format to the array. After the longest permitted integration time, the imager turns into the read-out mode. The row buffers are disabled and the image processor starts to work. First, the QTD processor will generate linear quadrant address which is then translated into Hilbert scan address by the Hilbert Scanner block. The address is decoded into "Row Select Signal (RSx)" and "Column Select Signal (CSx)". The selected pixel will drive the data bus and its value will be first quantized by the DPCM Adaptive Quantizer then the binary quantization result will be compressed by the QTD processor.

B. Hilbert Scanner

Hilbert scanning is found to be composed by multiple levels of four basic scanning patterns as shown in Fig. 5.

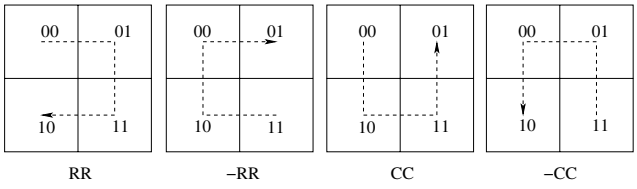


Fig. 5. Basic scanning patterns found in Hilbert scan.

They are denoted as RR, $-RR$, $-CC$, and CC respectively. For example, RR represents a basic scanning pattern featuring a relationship between its linear scanning sequence and physical scanning address as following:

RR: $('b00) \rightarrow ('b01) \rightarrow ('b11) \rightarrow ('b10)$,

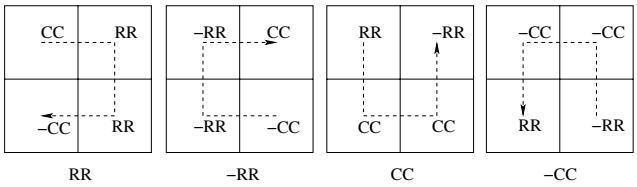
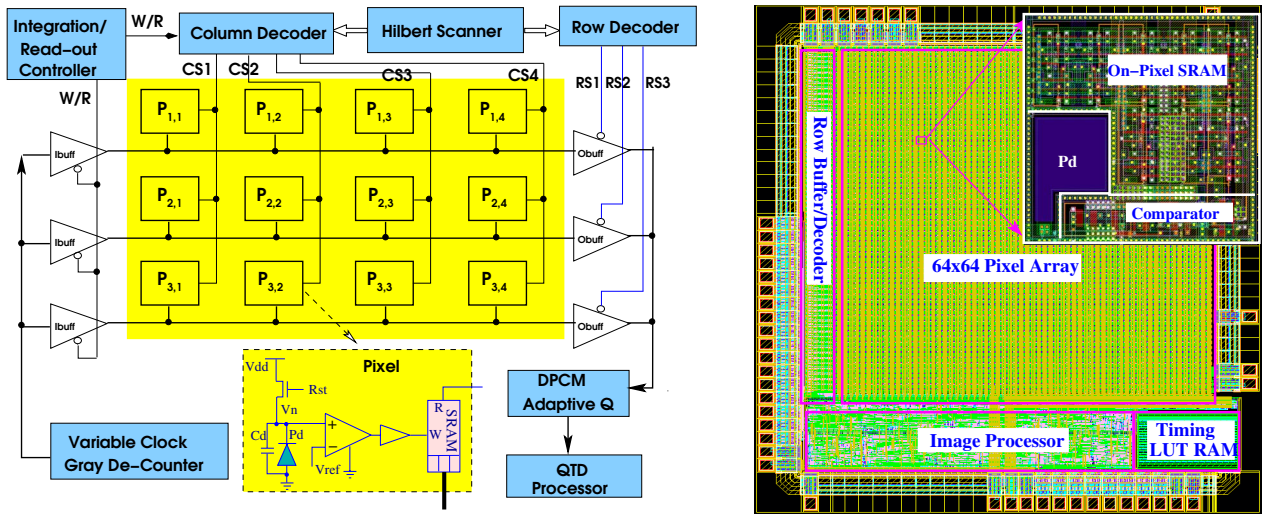


Fig. 6. Hierarchical Hilbert scanning sequence.

The whole array is then can be represented by hierarchies of such basic scanning sequences. If we look at some intermediate level, as shown by Fig. 6, if it is in the format of RR, then its four children quadrants must be in the format



(A.) (B.)

Fig. 4. A.) Imager Architecture B.) Imager layout implemented in Alcatel 0.35 μm CMOS technology with main blocks highlighted.

of $CC \mapsto RR \mapsto RR \mapsto -CC$. Based on this nice feature of Hilbert scanning, we can easily map the linear quadrant address into Hilbert scanning address using very simple hardware. First of all, we use a linear address to segment the whole array into levels of quadrants. Each level of quadrant is addressed by 2-bit of address. Then we map the linear address into Hilbert address in a top-down approach. At the highest quadrant level, the scanning sequence is predefined, can be RR or CC. Assume that we now using RR, then at the second highest level, the scanning type of the four quadrants are $CC \mapsto RR \mapsto RR \mapsto -CC$. Depending on the current linear quadrant address (first 2-bit of MSB), we are must within one of them. Let's further assume that we are now in the fourth quadrant which in the format of $-CC$. Then its four sub-quadrants must be in the format of $-RR \mapsto -CC \mapsto -CC \mapsto RR$. Further more, we can also determine which of them we are within now by looking at the second 2-bit MSB of the linear address. As a result, the whole process can be realized by a series of 2-bit address mapping without any sequential logic.

C. VLSI Implementation

The single chip image sensor and compression processor was implemented using $0.35\mu m$ AMI CMOS digital process (1-poly 5 metal layers). Fig. 4.(B) shows the chip's layout with a total silicon area of $3.2 \times 3.0mm^2$. The 64×64 pixel array was implemented using a full-custom approach. Each pixel occupies an area of $39 \times 39\mu m^2$ with a fill factor of 12%. The digital processing parts was synthesized from HDL and implemented using automatic placement and routing tools. The digital processor occupies an area of $0.25 \times 2.2mm^2$.

In Table II, we compared the number of flip-flops used in this processor compared to that reported in [4]. Significant saving is achieved mainly by two approaches: 1) storing the QTD tree information in the DPS array and 2) removing the boundary point storage by using Hilbert scan.

TABLE II
NO. OF FLIP-FLOPS USED IN THIS WORK AND [4].

Function Block	This work	That of [4]
Adaptive η	9	9
DPCM	24	0
SmoothMZ	0	64
QTD	202	1407
HilbertScan	0	0
Total	**235**	**1480**

V. CONCLUSION

In this paper, a single chip CMOS image sensor with a hybrid 1-bit FBAR quantizer combining DPCM algorithm and QTD compression processor is presented. Hilbert scan is employed to provide both spatial continuity and quadrant based scan. The whole processor is implemented with small hardware expense but achieves 0.75 BPP compression ratio and 28.5 dB image quality.

ACKNOWLEDGMENT

This work was supported by a grant from the Research Grant Council of Hong Kong (Ref.610405).

REFERENCES

[1] A. Olyaei and R. Genov, "Mixed-Signal Haar Wavelet Compression Image Architecture," *MWSCAS'05*,Cincinnati, Ohio, 2005.

[2] Kawahito, et *al.*, "CMOS Image Sensor with Analog 2-D DCT-Based Compression Circuits," *JSSC*, Vol.32, No.12, pp.2029-2039, Dec. 1997.

[3] L. G. Chen, et *al.*, "A lowpower 8×8 direct 2D-DCT chip design," in *J. VLSI Signal Process.*, Vol.26, pp.319-332, 2000.

[4] S. Chen, et *al.*, "Adaptive-Quantization Digital Image Sensor for Low-Power Image Compression," *TCASI*, Vol.54, Issue 1, pp.13-25, Jan. 2007.

[5] E. Artyomov, et *al.*, "Morton (Z) Scan Based Real-Time Variable Resolution CMOS Image Sensor," *IEEE Trans. On Circuits and Systems For Video Technology*, Vol.15, pp.947-952, Jul. 2005.

[6] Biswas, S., "Hilbert scan and image compression," *15th International Conference on Pattern Recognition*, Vol.3, pp.207-210, 2000.

[7] D. Martinez, et *al.*, "Generalized Boundary Adaptation Rule for Minimizing rth Power Law Distortion in High Resolution Quantization", *Neural Networks*, Vol.8, No.6, pp.891 -900, 1995.