A CMOS Image Sensor with combined adaptive-quantization and QTD-based on-chip compression processor

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*— In this paper, a CMOS image sensor with on-chip compression processor is proposed. An adaptive quantization scheme based on boundary adaptation procedure followed by an on-line quadrant tree decomposition processing is proposed enabling low power, robust and compact image compression processor. The image sensor chip has been implemented using $0.35\mu m$ CMOS technology and operates at 3.3 V. Simulation and experimental results show compression figures corresponding to 0.6-0.8 BPP, while maintaining reasonable PSNR levels and very low operating power consumption.**

I. Introduction

The aggressive scaling of CMOS technology has enabled the design of higher resolution and higher frame rate image sensors, featuring improved image and video quality but at the expense of increased output bandwidth. For portable wireless video sensors, this increased output data rate translates into higher transmission power dissipation, wider channel bandwidth and increased memory size. Image compression is therefore becoming increasingly important, particularly with the emergence of new Mega-pixels image sensors. A number of on-chip image compression implementations have been reported in the literature [1], [2]. Unfortunately, image compression remains the most expensive hardware in digital video camera [1], which even if implemented on-chip would result in high power consumption and large silicon area. This would limit the prospect of implementing low power image acquisition and image compression on a single chip.

In this paper, adaptive quantization scheme based on boundary adaptation procedure followed by an efficient on-line quadrant tree decomposition algorithm is proposed to achieve low power, compact and robust image compression integrated together with a digital CMOS image sensor. The image is first acquired using a non-destructive digital pixel sensor array followed by an adaptive quantization scheme that permits to compress the data to a lower number of bits (typically 1 to 2 bits per pixel). Further compression is accomplished, while scanning out the pixel values, using a lossless Quadrant Tree Decomposition (QTD) algorithm. QTD compresses spatially redundant data in the binary image and allows to achieve ≤ 1 bit per pixel, without any further degradation of the image quality. The remainder of the paper is organized as follows. Section II introduces the algorithmic considerations for both the adaptive quantization and the QTD compression algorithms. The compression performance expressed in terms of bit-per-pixel (BPP) and the image quality expressed in terms of PSNR are also reported in this section. Section III describes the VLSI architecture and the experimental results obtained from the prototype chip. Section IV concludes this paper.

II. Adaptive Quantization and QTD compression Algorithm

A. Backward adaptive quantization

The proposed adaptive quantizer can be specified by an ordered set of boundary points $x_0 < x_1 < \cdots < x_{i-1} < x_i < \cdots < x_{N-1} < x_N$ delimiting N disjoint quantization intervals $R_1, \cdots, R_i, \cdots, R_N$, with $R_i = [x_{i-1}, x_i]$. The size of the quantization interval i is noted by $\delta_i = (x_i - x_{i-1})$. The quantizer maps pixel intensity u_n sampled at time n into one of N quantization levels y_i, $i = 1 \cdots N$, such that

$$\hat{u}_n = \sum_{i=1}^{N} y_i \, \mathbb{1}_{R_i}(u_n) \tag{1}$$

with $\mathbb{1}_{R_i}(u_n) = 1$ if $u_n \in R_i$ and $= 0$ otherwise. The quantizer output I_n is defined by the N-bit binary vector $(\mathbb{1}_{R_1}, \cdots, \mathbb{1}_{R_N})$, although a more compact representation J_n is actually obtained by an additional processing stage for transmission. The reconstruction levels y_i are taken as the midpoints of their corresponding quantization intervals: $y_i = (x_{i-1}+x_i)/2$. The boundary points delimiting the quantization intervals are therefore the only parameters to adapt.

In our quantizer, the extreme boundary points x_0 and x_N are fixed by the quantization range but the other boundary points from x_1 to x_{N-1} are parameters that change over time. Because the decoder has a structure similar to the encoder, the same adaptation rule is implemented at both sides of the channel. At each time step n, the transmitted codeword J_n is used to adjust the quantizing parameters (backward adaptation)

$$\Delta x_i = x_i(n) - x_i(n-1) \tag{2}$$

where $i = 1 \cdots N-1$. The backward adaptation rule, called $FBAR_r$ for Fast Boundary Adaptation Rule, is obtained by updating all boundary points at each time step :

$$\Delta x_i = \frac{\eta}{N-i} \sum_{k=i+1}^{N} \delta_k^r \mathbb{1}_{R_k} - \frac{\eta}{i} \sum_{k=1}^{i} \delta_k^r \mathbb{1}_{R_k} \tag{3}$$

where η is the step size, a positive scalar. It has been shown that $FBAR_r$ (Eq. (3)) minimizes an r-th power law distortion D_r [3], 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 Eq. (3) becomes

$$\Delta x = \eta(\mathbb{1}_{R_2} - \mathbb{1}_{R_1}) \tag{4}$$

where x is the unique boundary point, R_1 and R_2 are the left and the right quantization intervals, respectively. At each time step, x is thus increased or decreased by $\Delta x = \pm\eta$. At convergence, we have on average $< \mathbb{1}_{R_2} >=< \mathbb{1}_{R_1} >$ and $< \Delta x >= 0$. The boundary point x oscillates around the median value of the input so that the probability of having either R_1 or R_2 active is $1/2$.

B. QTD compression

The adaptive quantizer explained earlier permits to build a binary image on which quadrant tree decomposition (QTD) is further employed to achieve higher compression ratio. The QTD compression algorithm is performed by building a multiple hierarchical layers of a tree, in which each node represents the compression possibility of a square block within the pixel array. The array is scanned following the Morton (Z) scan [4] order, which features simple address mapping relationship between pixel's tree address and its physical address within the array. While scanning, at each level of the tree, a comparator will compare the incoming code words with a number of θ inputs, where θ is dependent upon the tree level. For example, at the bottom level θ=4, while at the second bottom level θ=16. At the end of every θ clock cycles, the comparison result refereed to as flag bit is stored into a flip-flop. Flag bit values equal to "1" indicates that the corresponding quadrant within the array can be compressed as all its values are equal. The tree construction procedure described earlier appears as a bottom-up approach, however the procedure used is in fact performed in parallel.

C. Smooth Boundary Point Propagation

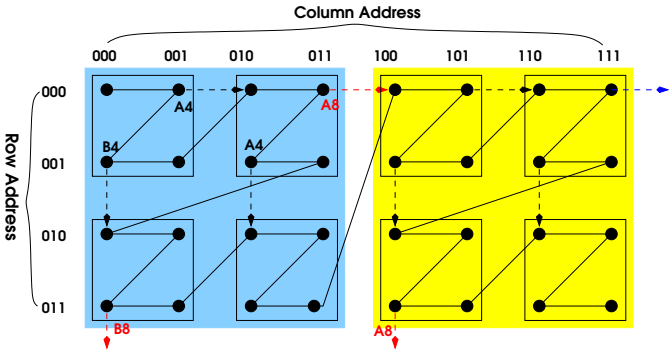

Fig. 1. Smooth boundary point propagation scheme for 4×4 and 8×8 pixel quadrants. Two registers $(A4, B4)$ are needed to store the boundary point for the 4×4 quadrant level and two other registers $(A8, B8)$ are needed to store those related to the 8×8 quadrant level.

The Morton (Z) scan strategy [4] is a quadrant or window-based read-out, which is compatible with QTD algorithm. A direct mapping is obtained between the QTD tree structure and the pixel array using odd and even addresses. Unfortunately, Morton (Z) scan presents a serious drawback when combined with the adaptive quantizer, presented earlier. The transition from one quadrant to the next involves jumping to a non-neighboring pixel resulting in spatial discontinuity affecting the performance of the adaptive quantizer. Due to the inherent hierarchical partition of the QTD algorithm, this transition gets larger and larger when scanning the array. As a consequence, one can expect sharp deviations in the pixel's values during transitions from one quadrant to another. This will introduce large errors in the adaptive quantizer at the edge of the quadrants. To address this problem, we propose a smooth boundary point (BP) propagation scheme, as shown in Fig. 1. One can note that when the Morton (Z) scan transits from one quadrant to another (discontinues BP), instead of taking the BP from the previously scanned pixel, the BP is taken from the physically nearest neighbor of the previous quadrant. For a 1-bit quantizer, a simple digital circuit is required for address detection while two registers are needed at each tree layer for BP storage. For a 2-bit quantizer, the number of BP involved is 3 times larger and hence 6 registers are needed to store the discontinues BP.

D. Simulation results

We have evaluated the performance of our proposed algorithm for a 1-bit and 2-bit adaptive quantizer followed by QTD. The performance is found to be dependent upon the choice of parameter η. On one hand, a large η is needed so as to track rapid fluctuations in consecutive pixel values. On the other hand, a small η is needed so as to avoid large amplitude oscillations at convergence. To circumvent this problem, 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 $\Lambda > 1$ ($\Lambda = 1.125$ here). 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 shows the simulation results obtained for Lena image using the proposed techniques. It is clear that a 2-bit quantizer features improved PSNR as compared to a 1-bit quantizer. Better performance in terms of both PSNR and compression ratios are obtained when using our smooth boundary Morton (Z) scan methodology as compared to a raster scan. It was found that the adaptive η not only provides an improved PSNR but also improved compression ratio. More simulations were performed on seven sample images and the results are illustrated in table I, which confirms our findings. Morton (Z) scan permits to hierarchically access square blocks of pixels presenting higher likelihood of similarity as it is a block-based read-out strategy. Besides, the adaptive η uses larger adaptation steps for fast transient in the original image, while small steps

TABLE I

PSNR (dB) AND BIT-PER-PIXEL (BPP) FOR SOME SAMPLE IMAGES USING 1-BIT Q WITH FIXED η RASTER SCAN (1-B η_0-R), 1-BIT Q WITH ADAPTIVE η RASTER SCAN (1-B η-R), 1-BIT Q USING FIXED η SMOOTH BOUNDARY MORTON (Z) SCAN (1-B η_0-MZ), 1-BIT Q USING ADAPTIVE η SMOOTH BOUNDARY MORTON (Z) SCAN (1-B η-MZ), 2-BIT Q USING RASTER SCAN (2-B η_0-R) AND 2-BIT Q USING SMOOTH BOUNDARY MORTON (Z) SCAN (2-B η_0-MZ).

Quantizer	Sample Images														Average		
	Lena		News		Cancer		Gut		Elaine		Plane		Bacteria				
	psnr	BPP	psnr	BPP	psnr	BPP	psnr	BPP	psnr	BPP	psnr	BPP	psnr	BPP	psnr	BPP	R
1-b η_0-R	22.9	0.88	22.9	0.58	26.2	0.76	29.0	0.70	26.0	0.87	28.8	0.50	28.7	0.73	26.4	0.71	37.18
1-b η-R	24.6	0.89	24.0	0.59	27.1	0.75	31.9	0.71	26.4	0.87	30.1	0.52	30.6	0.61	27.8	0.70	39.71
1-b η_0-MZ	27.1	0.72	25.9	0.56	28.5	0.69	32.2	0.58	28.5	0.73	30.1	0.49	32.6	0.73	29.2	0.64	45.62
1-b η-MZ	27.6	**0.71**	26.1	**0.56**	29.0	**0.68**	33.9	**0.57**	29.0	**0.71**	30.8	**0.50**	32.7	**0.61**	29.9	**0.62**	**48.22**
2-b η_0-R	25.4	1.93	25.2	1.28	29.5	1.77	31.0	1.63	27.5	1.95	32.5	1.05	32.9	1.57	29.0	1.59	18.24
2-b η_0-MZ	**29.9**	1.59	**28.4**	1.11	**31.0**	1.62	**34.0**	1.31	**30.2**	1.71	**33.8**	0.78	**33.9**	1.55	**31.6**	1.40	22.57

Fig. 2. Simulation results for Lena image. (A.) is the 256×256 original image. Figures (B.), (C.) and (D.) represent the results for 1-bit quantizer using fixed η raster scan, 1-bit quantizer using adaptive η smooth boundary Morton (Z) scan, 2-bit quantizer using smooth boundary Morton (Z) scan, respectively.

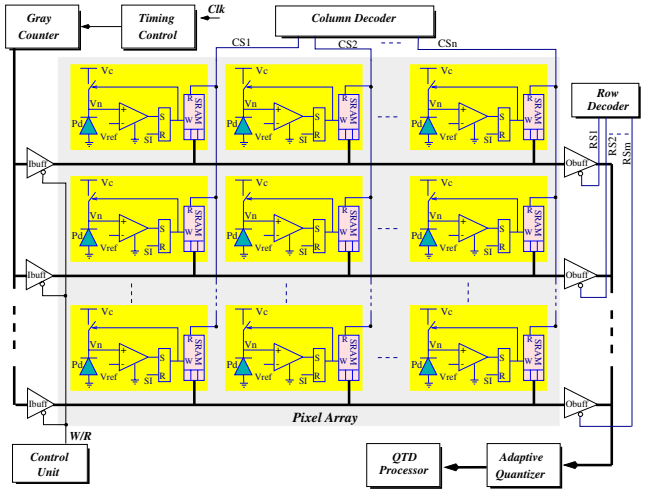

Fig. 3. Block diagram of a single chip CMOS image sensor with the adaptive quantizer and the QTD processor.

are used in the case of stationary or slow varying signals. This in turns increases the convergence speed and decreases the mismatch between the original image and its quantized counterpart.

III. VLSI ARCHITECTURE AND EXPERIMENTAL RESULTS

A. VLSI Architecture

The architecture of the CMOS image sensor and the proposed compression processor is shown in Fig 3. The image array consists of 64×64 digital pixel sensors equipped with pixel level non-destructive storage elements. Each pixel is composed of a photosensitive device (reverse biased photodiode P_d) with its internal capacitance C_d, a reset transistor, a comparator and a feedback circuit [5]. The voltage at the sensing node of the photodiode (V_n) is first reset to V_C. After the reset phase, the light falling onto the photodiode discharges C_d, resulting in a decreasing voltage V_n across the photodiode node. The accumulated charge in the pixel is converted to a time stamp using a comparator and an SR latch. A Gray counter is used in order to convert this timing stamp into a digital code stored in each memory. After the integration phase, the pixel array can be viewed as a distributed static memory and the adaptive quantization as well as the QTD compression are performed in parallel during the read-out phase of the array. Fig. 4 shows the diagram of the 1-bit adaptive quantizer (blocks within the solid line box), which includes a digital comparator $C1$, an 8-bit multiplexer $MUX1$, an 8-bit adder and one 8-bit register (BP Reg). As the pixel value is read from the array using a gray encoding, a gray-to-binary conversion is also required.

The adaptive quantizer compares the pixel value read out from the array with the current BP value, which is initially set to the mid-range. The boundary point is then adjusted by an adaptation step size ($\pm\eta$) depending upon the comparison result. A D flip-flop and a XOR gate are added in order to detect if two consecutive comparison results are equal. If this is the case, the value of η is increased by a ratio set to be 1.125, by selecting the right output of the multiplexer $Mux2$. The value of η is then adapted and used to adjust the boundary point. The same circuit can be extended for a higher number of bits.

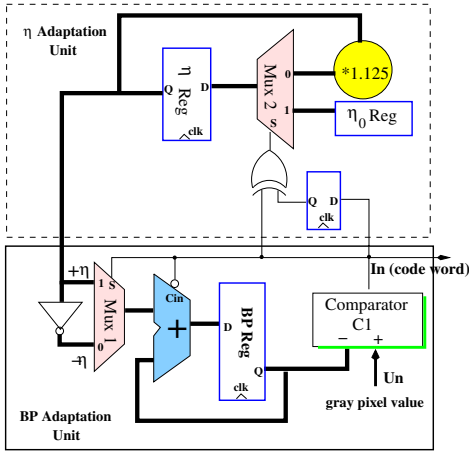

Fig. 4. 1-bit Adaptive quantizer building block. The building blocks represented under the solid line box correspond to the fixed η adaptive quantizer, while the building blocks represented under the dashed line box are the circuit extension required to realize the adaptive η quantizer.

B. Experimental results

The chip integrating the image sensor and the compression processor was implemented using $0.35\mu m$ AMI CMOS digital process. Fig. 5 shows the microphotograph of the chip with a total silicon area of $3.8 \times 4.5mm^2$. The digital processor which occupies an area of 1.8 mm^2 includes a large number of operating configurations such as: 1-bit and 2-bit quantizers with fixed and adaptive η, with and without QTD and using both raster and smooth boundary Morton (Z) scan.

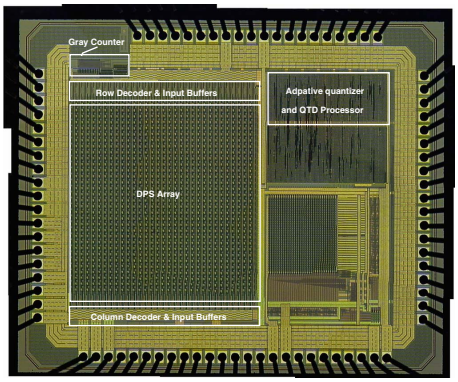

Fig. 5. Microphotograph of the prototype Chip.

The reconfigurable adaptive and fixed η 1-bit adaptive quantizer requires only $1.6K$ transistors and consumes less than $1mW$ estimated power while achieving compression ratios corresponding to less than $1BPP$. It is worthwhile to note that although QTD building block requires the largest number of transistors (46K mainly required for storing the flag bits), it still consumes little power (about $2mW$). This is explained by the hierarchical nature of the circuit with a maximum of $log_2 n$ cells (n=64 in our circuit) being updated during each iteration of the tree construction.

Sample 64×64 images were acquired from the prototype using different operating modes shown in Fig. 6. Visually, it is quite obvious that the 2-bit quantizer using smooth boundary point Morton (Z) scan presents the best image quality for all sample images. The 1-bit quantizer using adaptive η and smooth boundary Morton (Z) scan performs better in terms of image quality as compared to all 1-bit adaptive quantizers.

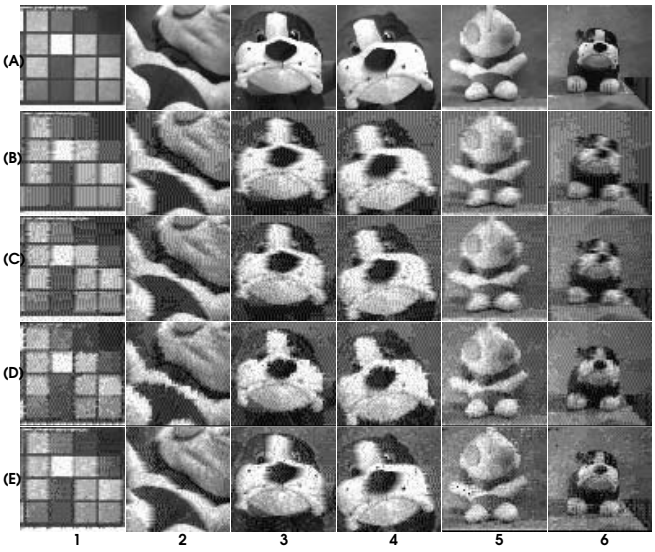

Fig. 6. Captured images under different processing modes. Row (A.) shows the 8-bit captured images without compression, (B.), (C.), (D.) and (E.) represent the reconstructed compressed images using 1-bit adaptive quantizer with fixed η raster scan, 1-bit adaptive quantizer with adaptive η raster scan, 1-bit adaptive quantizer using adaptive η smooth boundary Morton (Z) scan and 2-bit quantizer using smooth boundary Morton (Z) scan, respectively.

IV. Conclusion

A single chip CMOS image sensor and a compression processor is reported in this paper. A novel compression algorithm based on boundary adaptive quantization and an efficient on-line quadrant tree decomposition QTD is proposed. The performance in terms of both image quality (PSNR) and compression ratio (BPP) were further improved using a novel smooth boundary Morton (Z) scan and a heuristic adaptive boundary rule, which were also implemented in VLSI. Results showed that 0.6-0.8 BPP can be achieved while using a very compact and low power ($< 20mW$) 1-bit adaptive quantizer with smooth boundary Morton (Z) scan.

Acknowledgment

This work was supported by a Univ. grant and a grant from the RGC of Hong Kong (610405 and HIA05/06.EG03).

References

[1] Kawahito et *al.*, "CMOS Image Sensor with Analog 2-D DCT-Based Compression Circuits," *IEEE JSSC*, Vol. 32, pp.2029-2039, Dec. 1997.
[2] Q. Luo and J. G. Harris, "A novel Integration of on-sensor Wavelet Compression for a CMOS Imager," *ISCAS02*, Vol. III, pp.325-328, 2002.
[3] D. Martinez and M.M. Van Hulle, "Generalized Boundary Adaptation Rule for minimizing r-th power law distortion in the high resolution case," *Neural Networks*, 1995.
[4] 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. 05.
[5] A. Kitchen, et *al.*, "A DPS Array With Programmable Dynamic Range,", *IEEE TED,* Vol. 52,pp.2591-2601, Dec. 2005.