Smooth Boundary Point Adaptive Quantizer for on-chip Image Compression

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

***Abstract*— This paper proposed a smooth boundary point adaptive quantization scheme used for image compression. The proposed algorithm is first validated through Matlab simulation then implemented together with a CMOS image sensor. Simulation and experimental results show that compression figures corresponding to 0.6-0.8 bit-per-pixel are achieved while maintaining reasonable image quality.**

I. Introduction

CMOS image sensors are gaining a very large attention due to their inherent advantages of low power, low cost and the possibility for the realization of image acquisition as well as image processing on a single chip. 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 requiring extensive research work. In such applications, it is often required to capture a large amount of data and process them in real-time while the hardware is constrained to take very little physical space and to consume very little power. This is only possible using custom single chip solutions integrating image sensor and hardware-friendly image compression algorithms. 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.

Novel algorithms and VLSI architectures are therefore required. In this paper, we propose an adaptive quantization scheme integrated together with a digital TFS CMOS image sensor [4] based on boundary adaptation procedure which can compress the data to 1 bit-per-pixel (BPP). Further compression is achieved (0.6-0.8 BPP) using quadrant tree decomposition (QTD) algorithm. However, the Morton(Z) [5] scan of QTD algorithm presents a serious drawback when combined with the adaptive quantizer. 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. To address this problem, a smooth boundary point propagation scheme is proposed which features simple hardware implementation and proved to be efficient. Section II introduces the adaptive quantization algorithm followed by the smooth boundary point propagation scheme described in section III. Section IV shows the architecture and the experimental results of the prototype chip. Section V concludes this paper.

II. Backward Adaptive Quantization

The proposed quantizer is 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 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. The quantizer maps pixel intensity u_n sampled at time n into one of N quantization levels. At each time step n, the transmitted codeword is used to adjust the levels (backward adaptation[6]):

$$\Delta x_i = x_i(n) - x_i(n-1)$$

where $i = 1 \cdots N-1$. The backward adaptation rule, called $FBAR_r$ for Fast Boundary Adaptation Rule, can be explained by an example shown in Fig.1.

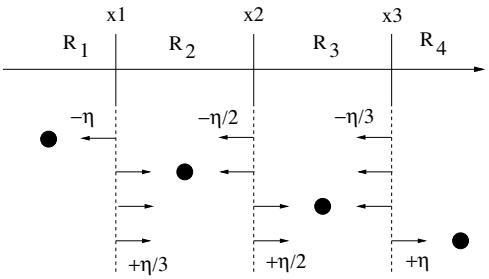

Fig. 1. Example of the backward adaptation $FBAR_0$ for a 2-bit quantizer.

There are three adaptive boundary points x_1, x_2 and x_3 delimiting four quantization intervals from R_1 to R_4. The reconstructed values are taken as the midpoint of their corresponding intervals, $y_2 = (x_1 + x_2)/2$ and $y_3 = (x_2 + x_3)/2$, except for the extreme intervals for witch $y_1 = x_1$ and $y_4 = x_3$. At each time step, the input pixel intensity falls within a given interval. Thus, there are four cases to be considered. For each one, the active interval is indicated by a black dot in the figure. Each boundary point is shifted in the direction of the active interval by a quantity η divided by the number of bins on this side. When the interval R_1 is active, the boundary points x_1, x_2 and x_3 decrease by $-\eta$, $-\eta/2$ and $-\eta/3$, respectively (see 1^{st} adaptation row in the figure). When R_2 is active, x_1 increases by $+\eta/3$ and x_2 and x_3 decrease by $-\eta/2$ and $-\eta/3$, respectively (see 2^{nd}

1-4244-0765-6/06/$20.00 ©2006 IEEE

adaptation row in the figure). When R_3 is active, x_1 and x_2 increase by $+\eta/3$ and $+\eta/2$ and x_3 decreases by $-\eta/3$ (see 3^{rd} adaptation row in the figure). When R_4 is active, x_1, x_2 and x_3 increase by $+\eta/3$, $+\eta/2$ and $+\eta$, respectively (see the last adaptation row in the figure).

The adaptation step size parameter η is found to affect the quantizer performance. 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.

III. Smooth Boundary Point Propagation

A. Operating Principle

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 scanning the array following the Morton(Z) [5] scan strategy to build a multiple hierarchical layers of a tree, in which each node represents the compression possibility of a square block within the pixel array. The Morton(Z) scan strategy is a quadrant or window-based read-out, which is compatible with QTD algorithm and features very simple hardware implementation. Unfortunately Morton(Z) scan presents a serious drawback when combined with the adaptive quantizer.

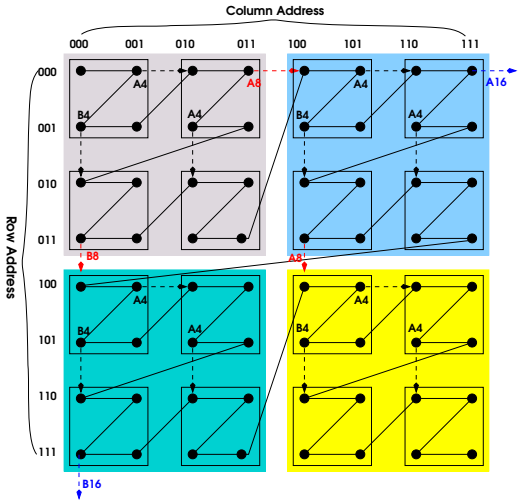

Fig. 2. Smooth boundary point propagation scheme for 4×4 and 8×8 pixel quadrants. Two registers $(A4, B4)$ are needed to store the boundary point for the 4×4 quadrant level and two other registers $(A8, B8)$ are needed to store those related to the 8×8 quadrant level.

As shown in Fig.2, 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. Actually the transition step size is dependent on the hierarchy level of the quadrant. 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 propagation scheme. In this scheme when the Morton(Z) scan transits from one quadrant to another, instead of taking the boundary point from the previously scanned pixel, the boundary point is taken from the physically nearest neighbor of the previous quadrant as shown in Fig.2.

B. Hardware Implementation

Implementing such a scheme is not very complicated as storing boundary points from specific locations is repeatedly required. At the quadrant level of 4×4, an address detector will monitor the scanning row and column address. Whenever the address meets the condition $rowaddr =' b00\&coladdr =' b01$ or $rowaddr =' b01\&coladdr =' b10$, a register $A4$ will be enabled to record the current boundary point (BP). The stored BP will later be loaded out for quantization when the address meets the condition $rowaddr =' b00\&coladdr =' b10$ or $rowaddr =' b10\&coladdr =' b10$. Similarly, another register $B4$ will be enabled to store the current BP when the address meets the condition $rowaddr =' b01\&coladdr =' b00$ and be loaded out at the address $rowaddr =' b10\&coladdr =' b00$. This smooth boundary point propagation scheme is illustrated by Fig.2. In the case of a 2-bit quantizer, 6 registers are required as the number of boundary points involved is 3 times larger as compared to 1-bit quantizer.

Fig.3 shows the architecture of the proposed compression processor with smooth BP propagation scheme. For the sake of clarity, Fig.3 only shows the hardware necessary for the 4×4 quadrant level smooth boundary point propagation. The adaptive quantizer compares the digital pixel value read out from the sensor 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. To enable the 1-bit quantizer with smooth boundary point propagation, additional write/read control circuits are built for the register $A4$ and $B4$. Two $And4$ and one $Or2$ gates are used as the address detector and will only enable the $A4$ register to record the current boundary point when the LSB of the address meets the condition: $rowaddr =' 00 and coladdr =' 01$ or $rowaddr =' 01 and coladdr =' 10$. Similarly, the $B4$ register will only be enabled to write the current BP at the address: $rowaddr =' 01 and coladdr =' 00$. The BP value stored in $A4$ register will only be loaded as one of the comparison operand when

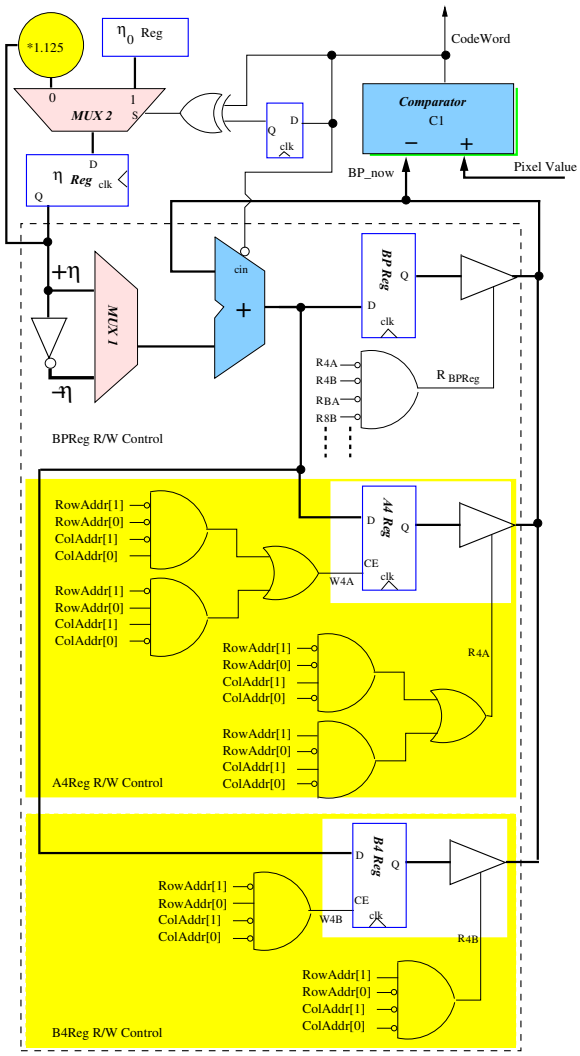

Fig. 3. 1-bit Adaptive quantizer building block. The building blocks inside the dashed line box correspond to the smooth boundary points write/load circutits. For the sake of clarity, only the two registers of 4×4 level is shown. The circuits outside of the box are the comparator and the circuit required to realize the adaptive η quantizer.

the address meets: $rowaddr =' 00 and coladdr =' 10$ or $rowaddr =' 10 and coladdr =' 10$. The value stored in the conventional BP_{reg} has the lowest priority and its value will be read out only when the $A4$, $B4$, $A8$, $B8$... registers are not selected. The write/read circuits for $A8$, $B8$, $A16$... are also as simple as that of the $A4$ and $B4$ registers. The only difference is that address detector will monitor more bits of address.

C. Simulation results

The proposed techniques are validated through extensive Matlab simulation. In order to evaluate the effect of using the Morton(Z) scanning procedure as compared to a conventional raster scanning, we also compared the PSNR using both scanning methodologies. As shown in Fig.4B and Fig.4C, it is clear that better performance is achieved using Morton(Z). One can see that the Fig.4B is blurred at the edges of the face and hair because the fixed η and raster scanner cannot catch up with rapid fluctuations at the edges of the image. Fig.4(C.) gives sharper edges due to the use of adaptive η and Morton(Z) scan which permits to hierarchically access square blocks of pixels presenting higher likelihood of similarity. It is however clearly noticed that the image in Fig. 4(C.) presents a number of noisy points introduced from the hierarchically propagated boundary points. The quantizer cannot adapt to the sharp deviations when transition between the quadrant occurs and large quantization error is observed. This issue is well addressed using the smooth boundary point propagation scheme as shown in Fig.4D.

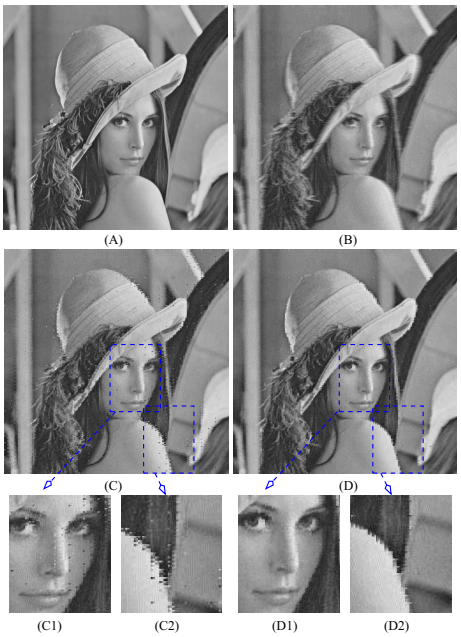

Fig. 4. Simulation results for Lena image. (A.) is the 256×256 original image. (B.) represents the results for 1-bit quantizer using fixed η raster scan, Figures (C.) and (D.) represent the 1-bit quantizer using adaptive η Morton(Z) scan without and with smooth boundary, respectively.

IV. Imager Architecture and Experimental Results

A. Imager Architecture

The architecture of the single chip image sensor and compression processor is illustrated in Fig.5. The image array consists of 64×64 digital pixel sensors equipped with pixel level 8-bit SRAM. The voltage at the sensing node of the photodiode (V_n) is first reset to $V_C = V_{dd} - V_{TH}$. 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 time required for V_n to reach the threshold voltage V_{Ref} of the comparator and hence to generate the SRAM write enable signal W_{en} can be interpreted as the time-to-first spike. A time stamp provided by a global timing unit and gray counter is therefore recorded by the on-pixel SRAM. Once the integration phase is completed, 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.

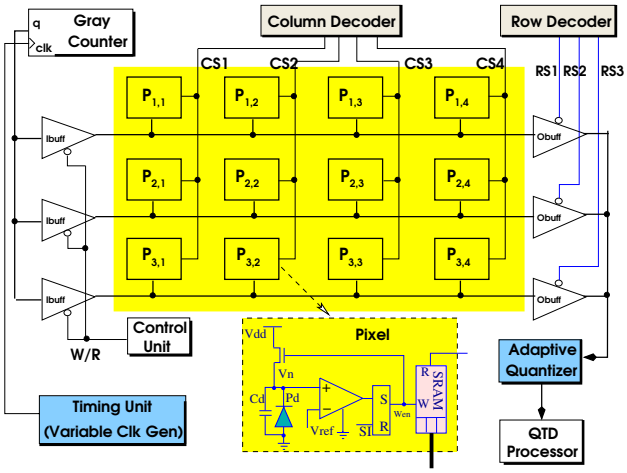

Fig. 5. Block diagram of a single chip CMOS image sensor with the adaptive quantizer and the QTD processor.

B. Experimental results

The single chip image sensor and compression processor was implemented using $0.35\mu m$ AMI CMOS technology occupying a silicon area of $3.8\times4.5mm^2$. The pixel array was implemented using a full-custom approach while the digital processing parts related to smooth adaptive quantizer and the QTD compression was realized using automatic placement and routing tools. 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. One should note that if only a 1-bit or 2-bit quantizer followed by QTD processing is used without additional operating modes, this figure can be increased to more than 90% allowing to have most of the silicon area dedicated to the pixel array.

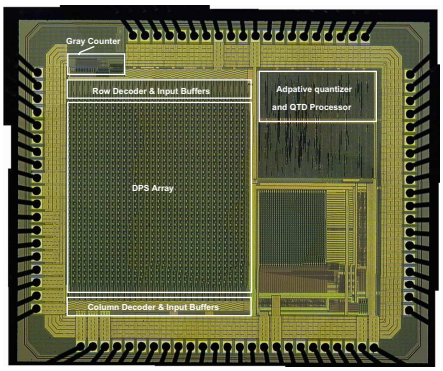

Fig. 6. Microphotograph of the prototype Chip.

Fig.6 shows the microphotograph with the main building blocks highlighted. Table I summarizes the performance of the imager. Sample 64 x 64 images were acquired from the prototype using different operating modes, as shown in Fig.7.

The 1-bit quantizer using adaptive η and smooth boundary Morton(Z) scan performs better in terms of image quality and compression ratio as compared to all other 1-bit adaptive quantizers. For example, the BPP values of rows (A), (B) and (C) of Fig.7 are: 8, 0.9, 0.8, respectively.

TABLE I
Summary of the imager performance

Technology	Alcatel 0.35μm, 3.3V
$V_{dd} - V_{Ref}$	0.5-0.8V
Pixel dynamic operating range	>100dB
FPN	0.8%
Pixel Area	45μm $\times$ 45μm
Pixel Array Area	>75%
Quantizer Transistors	53k
Quantizer Power Consumption	6.3mw

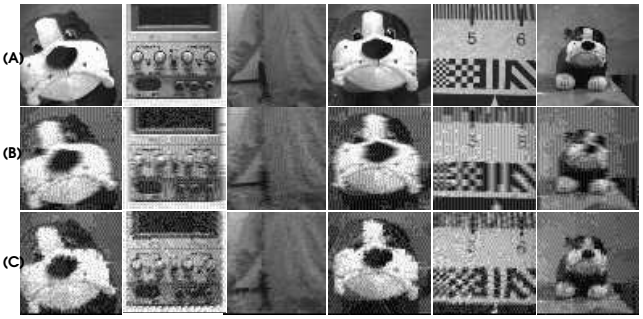

Fig. 7. Captured images under different processing modes. Row (A.) shows the 8-bit captured images without compression, (B.) and (C.) represent the reconstructed compressed images using 1-bit adaptive quantizer with fixed η raster scan, 1-bit adaptive quantizer using adaptive η smooth boundary Morton(Z) scan, respectively.

V. Conclusion

In this paper, a single chip CMOS image sensor with a boundary point adaptation algorithm and QTD compression processor is presented. A novel smooth boundary point propagation scheme is proposed to address the boundary point non-continuity problem associated to the Morton(Z) scan. Results showed that 0.6-0.8 BPP can be achieved while using a very compact silicon area and low power consumption.

Acknowledgment

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

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] A. Kitchen, et *al.*, "A DPS Array With Programmable Dynamic Range,", *IEEE TED,* Vol. 52,pp.2591-2601, Dec. 2005.
[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] D. Martinez et *al.*, "Generalized Boundary Adaptation Rule for minimizing r-th power law distortion in the high resolution case," *Neural Networks*, 1995.