

A NOVEL INTEGRATION OF ON-SENSOR WAVELET COMPRESSION FOR A CMOS IMAGER

Qiang Luo John G. Harris
Computational Neuro-Engineering Laboratory
University of Florida
453 Engineering Bldg., Bldg. 33
Gainesville, FL 32611-6130

ABSTRACT

A novel integration of image compression and sensing is proposed to enhance the performance of a CMOS image sensor. By integrating a compression function onto the sensor focal plane, the image signal to be read out can be significantly reduced and consequently the pixel rate can be increased. This can be applied to overcome the communication bottleneck for high-resolution image sensing and high frame-rate image sensing or for power- and bandwidth-constrained devices such as cell phones. A modified Haar wavelet transform is implemented as the compression scheme. A simple but efficient computation design is developed to implement the transform on-chip.

1. INTRODUCTION

CMOS imagers have been well studied since 1990 [1,2]. As pixel rates increase, it becomes more difficult for conventional image sensors to read out signals. The high bandwidth required to transfer data from the sensor, in the scan-and-read-out process, is a fundamental limitation for very high pixel rate imaging such as very high frame rate imaging or very high resolution imaging.

From the point of view of biological vision, integration of processing and sensing is reasonable in order to enhance sensing performance. Combining image acquisition and signal processing is an approach to overcome the communication bottleneck. Compression is considered an extremely desirable capability for an imager.

There has been work on the integration of sensing and processing in the implementation of neural networks and machine vision [3,4]. These implementations exploit the parallel nature of the image signal and result in fast processing, but the majority of these methods are focused on the implementation of specialized early-vision processing using resistive networks. Computational image sensors for on-sensor compression have also been reported [5,6]. For instance, a conditional replenishment, image compression algorithm is used in a sensor intended for video application but it suffers from low fill factor and bad fixed pattern noise (FPN). A CMOS imager with an analog two-dimension DCT-based compression [7] was reported

by Shoji Kawahito et al. To compute an 8×8 -point 2-D DCT, an eight-point data vector is read out eight times from an image sensor block to a unique DCT core. Slow processing reduces many of the benefits of the compression function.

In this paper, we investigate a novel integration of an image sensor and two-dimensional wavelet compression on the sensor plane. By integrating sensing and compression, the data bandwidth off the chip can be tremendously reduced. Furthermore, the most computationally intensive part of the wavelet transform is accomplished on the sensor plane, which significantly attenuates the load on downstream circuits. The algorithm embedded into the architecture is based on a well-known instance of the wavelet transform: the Haar transform. After deriving the formulation for a decimation of the two-dimensional Haar transform, an efficient transform scheme is proposed which requires only two computation units: a 2×2 element two-dimensional Haar transform unit and an adder or subtractor. A column-parallel architecture for processing and readout is implemented. The pixels in every N (N is a value of power of 2) rows are pulled down to N rows of switched-capacitor circuits simultaneously, where the transform is performed in an $N \times N$ block. The coefficients of the transform results are then compared with a threshold voltage. Only those coefficients that have values greater than the threshold are read out. A prototype chip with a 4×4 Haar transform block has been designed and fabricated using AMI $0.5 \mu\text{m}$ technology through MOSIS. Details of the modified Haar wavelet transform are presented in section 2. The circuit design is overviewed in section 3. Section 4 presents the simulation and experiment results.

2. MODIFIED TWO-DIMENSIONAL HAAR TRANSFORM

A simple but efficient wavelet transform used for image compression is the Haar transform, which has been widely used [8-12]. This section presents a modified Haar transform that is suitable for the on-sensor compression for a CMOS imager.

The forward discrete wavelet transformation (FDWT) transforms the DC-level shifted tile component samples $I(x,$

y) into a set of sub-bands with coefficients $w(u, v)$, which depend on the number of iterations (m), or the number of decomposition levels (Figure 1). The total number of sub-bands is $(3 \times m) + 1$. The sub-bands are represented in the form of “nXX”, where n denotes the decomposition level and “X” will be replaced by the letter “L” (low-pass filtering) or “H” (high-pass filtering). For example “1LL” means that the transform coefficients from the sub-band “1LL” are obtained from low-pass filtering vertically and horizontally at decomposition level 1. As illustrated in Figure 1, all the sub-bands in the case where $m=2$ can be represented in the following way:

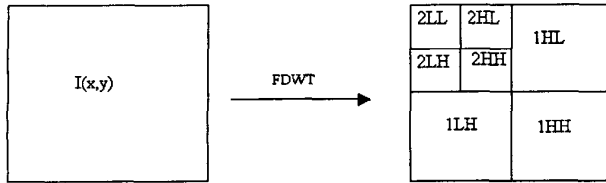


Figure 1. The decomposition of FDWT.

In practice, an $(M \times M)$ image is divided into an integer number of non-overlapping blocks, each of size $(N \times N)$, which are mapped into a 2D $N \times N$ Haar transform. Of the $N \times N$ Haar transform coefficients, only a few are significant. These are selected for quantization and the rest are set to zero.

Assuming an $N \times N$ pixel array, $\log_2 N$ iterations are needed to complete the Haar transform as discussed above. After each iteration, a scale of $1/2$ is applied to modify the resulting sub-band. Therefore, the maximum value of the transformed array (located at the left-top element) is guaranteed to be within the range of the image intensity. Furthermore, by scaling the transform coefficients by $1/2$, a simple arithmetic implementation of average is achieved for free (average of sum or average of difference). This function can be efficiently realized using capacitors and switches as discussed in the next section. Based on the above discussion, a basic transform scheme is proposed to realize the modified Haar transform. Equation group 1 presents the computation for a basic transform unit.

$$\begin{aligned} \beta_{11} &= \frac{1}{2} \left(\frac{\alpha_{11} + \alpha_{12} + \alpha_{21} + \alpha_{22}}{2} \right) \\ \beta_{21} &= \frac{1}{2} \left(\frac{\alpha_{11} + \alpha_{12} - \alpha_{21} + \alpha_{22}}{2} \right) \\ \beta_{12} &= \frac{1}{2} \left(\frac{\alpha_{11} - \alpha_{12} + \alpha_{21} - \alpha_{22}}{2} \right) \\ \beta_{22} &= \frac{1}{2} \left(\frac{\alpha_{11} - \alpha_{12} - \alpha_{21} - \alpha_{22}}{2} \right) \end{aligned} \quad (1)$$

where the original unit is a 2×2 array, named α , which is transformed to an array β through the modified Haar

transform. The subscripts indicate the elements of the arrays. Only adder and subtractor elements are needed and the calculation of dividing 2 can be treated as average of sum or difference, which is realized through charge redistribution in the circuit design. Therefore, the elements in the original image array are grouped into 2×2 blocks. For an $N \times N$ array, there are $\frac{N^2}{4}$ such blocks. Each block

processes the calculation characterized by equation group (1). Figure 2 shows an example of the transform on an 8×8 array (the original array is named A for convenience). The transformed array of A (named A1) overlaps the original array with rearranged element positions as shown in Figure 2(a). Those numbers in the figure, which represent the index of row and column, illustrate how the elements in the array A1 are located in array A. For example, A1(6,6) is located at the position of A(4,4). The highlighted elements are the elements that have been processed with the “LL” process (see Figure 1) and will be processed in the next iteration. These highlighted elements in A1 form a 4×4 sub-array (named the B array). The numbers in figure 2(b) show the rearranged element positions of the transformed array of B (named B1) on array B. The highlighted elements in B1 form a 2×2 sub-array and will be processed in the third (the last) iteration.

11	15	12	16	13	17	14	18
51	55	52	56	53	57	54	58
21	25	22	26	23	27	24	28
61	65	62	66	63	67	64	68
31	35	32	36	33	37	34	38
71	75	72	76	73	77	74	78
41	45	42	46	43	47	44	48
81	85	82	86	83	87	84	88

(a)

11	13	12	14
31	33	32	34
21	23	22	24
41	43	42	44

(b)

Figure 2 An example of the transform on an 8×8 array. The numbers in each small block represent the position of the element. The left number denotes row index, and the right number denotes column index. (a) represents the mapping of array A1 to array A. Those numbers illustrate how the elements in array A1 are located in array A. For example, A1(6, 6) is at the position of A(4, 4). (b) represents the mapping of transformed array of B (array B1) to array B. The highlighted elements are those processed with “LL” processing.

The elements processed in “LL” processing are scaled by 0.5 in each iteration until the end of the transform, but others are not. This unbalanced situation can be solved by simply scaling the threshold voltage of each element with the same scale value that is used in the transform.

3. DESIGN OVERVIEW

The CMOS imager contains a photodiode-type active pixel sensor (APS) array. Each pixel consists of a photodiode, a transistor working as a source follower, a reset transistor and a row-select transistor. This structure is similar to those demonstrated in the literature [13-17]. At the bottom of each column of the pixel array is a network of capacitors and switches that perform the proposed transform. In order to suppress the effect of fixed pattern noise (FPN), pixel KTC noise, and $1/f$ noise, a correlated double sampling (CDS) circuit is applied under each column. A row decoder at the side of the pixel array selects N (power of 2) rows of pixels each time and sends their values down to the CDS circuits and then output to the transform circuits for transform processing. The transform coefficients in each $N \times N$ block are then compared with a threshold voltage. Those coefficients whose value is greater than the threshold will be sent off-chip along with their position in the block.

Row pixel data is read onto the CDS circuit and then sampled onto two parallel capacitors that have the same capacitance (Figure 3). This sampled value is DC shifted and referred to half of the power supply voltage instead of ground. Shown in Figure 3(a) is the column parallel Haar transform unit circuitry, which consists of four elements in a 2×2 array. Pixel values are pulled down to the CDS circuits. The output of the CDS circuit is sampled to two identical capacitors so that two copies of the output are obtained. All 8 capacitors have the same capacitance. The capacitor bank and the switch bank together fulfill the transform scheme function. CLK1 switches realize the average of the sum or difference of two sampled pixel voltages as shown in figure 3. The CLK2 switches realize the average of the sum or difference of the post-CLK1 results (also see figure 3). The “w” (write) switches connect the sampling capacitors to the CDS circuitry outputs. The “res” (reset) switches reset the capacitors in the beginning of each processing circle. The “ref”, “ref1”, “ref2” and “ref3” switches connect the capacitors to the reference voltage. Three of the capacitors, as shown in Figure 3 (a), require special timings (named as inv1, inv2 and inv3) in the processing, which satisfy

$$\begin{aligned} \text{inv1} &= \overline{\text{res} + \text{ref1}} \\ \text{inv2} &= \overline{\text{res} + \text{ref2}} \\ \text{inv3} &= \overline{\text{res} + \text{CLK1}} \end{aligned}$$

The digital patterns shown in Figure 3(b) are the timing for the transform scheme unit. In this case, the “ref” switches are always closed. When this basic unit is involved in the

$N \times N$ transform processing, the timing becomes more complicated.

The analog signal processing using switched-capacitor circuits in CMOS image sensors is also reported in the work for multiresolution image sensors [18], where average of sum of voltages is implemented. The proposed method in this paper not only includes the processing for average of sum or difference of voltages, but also provides more computation power. Thus, more complicated processing can be achieved.

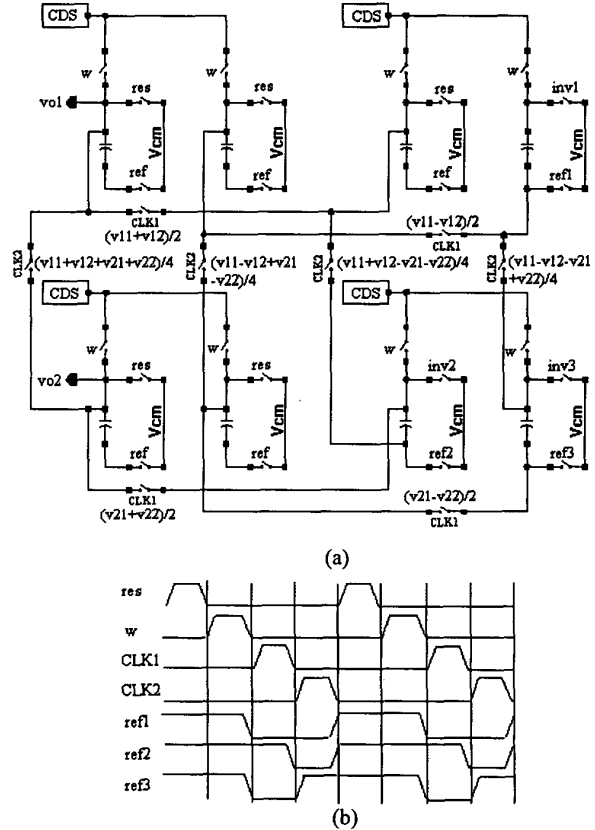


Figure 3 (a) schematic of the column parallel basic Haar transform unit circuitry. (b) the timing of the circuit.

4. SIMULATION AND EXPERIMENT RESULTS

Average of sum and difference are the two basic processing implemented in the transform. An example of the addition and subtraction of one transform process in CLK1 is illustrated in Figure 4. One of the input signals is set to be 2.5V ($V_{dd} = 5V$), the other one is swung from 0.5V to 3.5V. Figure 4(a) represents the average of sum processing results. The theoretic and simulated results are well matched. It shows that the accuracy is within 0.0056% (-85dB) of the ideal value. Figure 4(b) shows the average of

difference processing results. The accuracy decrease for the average of difference is due to the switch of polarity of capacitors. In practical implementation, the pixel saturation voltage limits the maximum voltage to be about 3.5V. The accuracy is within -39dB of the ideal value.

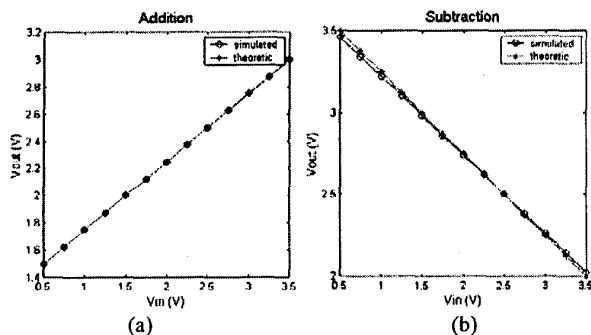


Figure 4 Accuracy testing for average of sum and difference.

A prototype chip of a 4×4 modified Haar transform array with CDS circuit for on sensor compression has been fabricated using AMI $0.5\mu\text{m}$ technology through MOSIS (see Figure 5). Chip test results show that the computation function can be fulfilled correctly. Unfortunately, due to a mistake in the design of the pad frame, GND was not provided to input buffer of a few input pads, which results in the miss of a few clock signals and we cannot get the final test results. A revised chip will be fabricated and further test will be done and reported later.

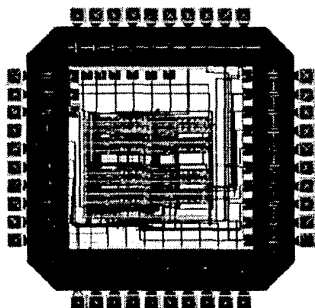


Figure 5 The prototype chip fabricated in AMI $0.5\mu\text{m}$ technology

5. CONCLUSION

In this paper, a novel integration of imaging and wavelet compression on an image sensor was proposed. The on-sensor compression is aimed at enhancing imaging performance of an image sensor. A modified simple but efficient calculation scheme for Haar wavelet transform was proposed to implement the on-chip column parallel compression. The proposed scheme is composed of average of sum or difference, and can be efficiently

fulfilled by switched-capacitor circuits. A prototype tiny chip is fabricated and tested.

REFERENCE

- [1] E. R. Fossum, "CMOS Image Sensors: Electronic Camera-On-A-Chip", IEEE Trans. on Electron Devices, vol. 44, no.10, pp1689-1698, Oct. 1997,
- [2] H. P. Wong, et al, "Single-Chip CMOS Imaging Systems", 1999 ISSCC Tutorial.
- [3] C. Koch, H. Li Eds, "Vision Chips: Implementing Vision Algorithms with Analog VLSI", Piscataway, NJ: IEEE Press, 1995.
- [4] J. G. Harris, et al, "Resistive Fuses: Analog Hardware for Detecting Discontinuities in Early Vision", In Analog VLSI Implementation of Neural Systems, Edited by Carver Mead and Mohammed Ismail, pp27-56, 1989
- [5] K. Aizawa, et al, "Computational Image Sensor for On Sensor Compression", IEEE Trans. On Electron Devices, vol. 44, no. 10, pp1724-1730, Oct 1997,
- [6] K. Aizawa, et al, "On Sensor Image Compression", IEEE Trans. On Circuits and Systems for Video Technology, vol 7, no. 3, pp543-548, June 1997.
- [7] S. Kawahito, et al, "A CMOS Image Sensor with Analog Two-Dimensional DCT-Based Compression Circuits for One-Chip Cameras", IEEE Journal of Solid-State Circuits, vol. 32, no. 12, pp 2030-2040, Dec 1997.
- [8] S. Mallat, "A Wavelet Tour of Signal Processing (2nd Ed)", Academic Press, 1999
- [9] A. M. Buron, et al, "Single-chip Fast Haar Transform at MHz Rates", In proc. 3rd Int. Workshop Spectral Technology, Dortmund, Germany, pp8-17, Oct. 1988.
- [10] K. J. R. Liu, "VLSI Computing Architectures for Haar Transform", Electron Letters, vol.26, no.23, pp1962-1963, Nov. 1990,
- [11] M. G. Albanesi, M. Ferreti, "A High Speed Haar Transform Implementation", J. Circuits, System Comput., vol. 2, no. 3, pp207-226, 1992.
- [12] G. A. Ruiz, J. A. Michell, "Memory Efficient Programmable Processor Chip for Inverse Haar Transform", IEEE Trans. Signal Processing, vol. 46, no. 1, pp263-268, Jan. 1998
- [13] S. Mendis, et al. "CMOS active pixel image sensor", IEEE Trans. On Electron Devices, vol.41, no.3, pp452-453, 1994.
- [14] S. Mendis, et al, "Progress in CMOS Active Pixel Image Sensors", In Charge-Coupled Devices and Solid State Optical Sensors IV, Proc. SPIE, vol.2172, pp19-29, 1994
- [15] D. Scheffer, et al, "Random Addressable 2048×2048 Active Pixel Image Sensor", IEEE Trans. On Electron Devices, vol.44 no.10, pp 1716-1720, Oct 1997.
- [16] G. Meynants, et al, "CMOS active pixel image sensor with CCD performance", In SPIE/EUROPTO AFPAEC conference, Zurich, Switzerland, May 18-19, 1998
- [17] R. H. Nixon, et al, "128x128 CMOS photodiode-type active pixel sensor with on-chip timing, control and signal chain electronics", In Proc. SPIE, vol. 2415, pp117-123.
- [18] S. E. Kemeny, et al. Multiresolution Image Sensor. IEEE Trans. On Circuits and Systems for Video Technology, vo. 7, no. 4, pp575-582, August 1997