Wavelet Transform Image Compression Prototype

Lanier Watkins, Kenneth R. Perry, John S. Hurley, Clark Atlanta University 223 J.P. Brawley Dr. SW Atlanta, GA 30304
B. Olson and B. Pain, Jet Propulsion Laboratory-California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109

ABSTRACT

In this study, we seek to develop a low power, area efficient wavelet compression chip capable of reconstructing sharp images at acceptable noise levels. It can be used in conjunction with such devices as the 256 x 256 CMOS Active Pixel Sensor (APS) camera under developed at JPL [1], or because of its small size, incorporated on the image sensor itself. A software algorithm is used to simulate the hardware and yield predicted values prior to fabrication. We limit our focus to the two-coefficient Haar Wavelet and one level Subband Coding (SBC). These parameters allow us to best emulate hardware restrictions in software. As a result, the software algorithm should yield very close findings to those of the hardware. Zerotree Encoding, which is a less restrictive algorithm, is employed as a standard. Reported results from Zerotree Encoding for 8:1 compression of the 512 x 512 standard Lena image yield a Peak Signal-to-Noise Ratio (PSNR) of 43.3 dB. Our software results for 8:1 compression of the 256 x 256 Lena image yield a PSNR of 37 dB, which is quite good given the more restrictive nature of our algorithm.

Keywords: wavelet transform, CMOS APS camera, low power, Subband Band Coding, prototyping

INTRODUCTION

Historically, image-processing methodologies have traditionally used the Discrete Cosine Transform (DCT) to accomplish synthesis and compression. However, recent efforts, including those of Watkins [2], Murenzi [3], and Namuduri [4] have shown that wavelet transform algorithms (continuous and discrete) may provide significant improvements over previously used algorithms. The software and hardware implementations used in this study are based on the use of the 2-coefficient Haar Wavelet and one level Subband Coding (SBC) [5]. This particular algorithm was chosen, because it is easily and accurately implemented in hardware. Zerotree Encoding, which is a less restrictive algorithm, utilizes the 9-coefficient Haar Wavelet and multi level SBC [6]. The Zerotree Encoding algorithm for 8:1 compression of the 512 x 512 Lena image yields a PSNR of 43.3 dB. In this effort the 256 x 256 Lena image, at 8:1, 4:1, and 2:1 compression ratios yield a PSNR of 37 dB, 37.3 dB, and 39 dB, respectively. Hence, the 2-coefficient Haar Wavelet and 1-level SBC compares favorably with other less restrictive algorithms. The chip is being designed for implementation in a NASA-JPL Active Pixel Sensor camera for image compression. Although it is not required for the chip to achieve very large compression ratios, minimum errors due to compression must be addressed. The results obtained from the software simulation reflect the probability that the proposed chip can indeed produce the necessary sharply reconstructed images with minimum errors in spite of the many restrictions introduced by the hardware. Ultimately, the completed prototype must undergo tests to measure performance and design efficiency. Our approach requires that such standard measurement tools as Root Mean Squared Error (RMSE) and Peak Signal-to-Noise Ratio[1] (PSNR) be used. The software algorithm will provide standard values that we can use to measure the performance and design efficiency of the chip.

Discrete Wavelet Transform (Software)

The Discrete Wavelet Transform has been implemented using Matlab. This implementation, shown in Figure 1 is based on Mallat's Standard Wavelet Transform Algorithm [5]. Mallat's algorithm uses quadrature mirror filters to provide wavelet analysis of an image. Large-scale analysis of the image is captured in the low frequency filtering, while small-scale analysis is captured in the high frequency filtering. The 2-coefficient Haar Wavelet consisting of both a lowpass filter [1 1] and a highpass filter [-1 1] is the preferred wavelet for this study because it can be readily implemented in hardware. The convolved lowpass filter and image produces the approximate or averaged coefficients while the convolved highpass filter and image produces detailed coefficients. The highpass and lowpass filters are called the decomposition filters because they break the image down or decompose the image into detailed and averaged coefficients, respectively. Similarly, the reconstruction lowpass and highpass filters [1 1] and [-1 1] respectively, can be used to rebuild the original image or to construct the wavelet function. The Matlab programming environment provided all the necessary tools needed to produce a menu driven software application. The image is read in as a 256 x 256 matrix and each row is convolved with a lowpass filter. The results are then stored in temporary matrix 1. The original image is then convolved with a highpass filter and stored in temporary matrix 2. Next, the columns of temporary matrices 1 and 2 are downsampled and stored into temporary matrices 3 and 4, respectively. The next step involves convolving the columns of temporary matrix 3 with lowpass and highpass filters, then storing the results into matrices 5 and 6, respectively. At this point, the columns of temporary matrix 4 are also convolved with lowpass and highpass filters. The results are stored in temporary matrices 7 and 8, respectively. Finally, the rows of temporary matrices 5-8 are downsampled and stored as subbands Lowpass Lowpass, Lowpass Highpass, Highpass Lowpass and Highpass Highpass, respectively. This procedure defines 1-Level Subband Coding using Mallat's Algorithm. These four subbands can be recombined to produce the original image if and only if none of the subbands are quantized. There exist many different types of quantization schemes, each with its own

This work was supported in part by the NASA Jet Propulsion Laboratory (Contract No. 961072).

characteristics and complexity. Unfortunately, hardware limitations restrict the allowable quantization schemes.

Discrete Wavelet Transform (Hardware)

Performing on-chip data reduction operations on the image sensor itself reduces the amount of power required for transmitting the data off-chip as well as subsequent hardware resources in the overall system. In our study we have developed a low power area efficient approach capable of being incorporated on the image sensor itself. Hardware implementation can be accomplished using essentially the same principles as the software implementation. However, a more efficient storage scheme is required to perform the operations accurately utilizing a reasonable amount of chip area.. In our study we assume a one-chip solution based on CMOS APS imager technology. Our approach could however be realized as a support chip and used in conjunction with a non-CMOS image sensor such as a CCD.

There are two parts to the chip, the imager array and wavelet compression module. The imager captures the image, and serves as input into the wavelet compression module. The pixels within the imager each contain an amplifier that converts the photo-generated charge at the sense node into a linearly proportional voltage. The amplifier in each pixel is connected through a switch onto a column bus. When the switch is closed, the detected voltage value is sampled onto a capacitor located at the bottom of the imager array. An entire row of image data can be obtained by broadcasting this control signal to all pixels in a given row of the imager. In this case there would have to be a column bus for every column of pixels in the imager and an equal number of capacitors located underneath the imager. In our approach we assume that there are two column buses for every column of pixels. In this way two rows of pixels can be read out simultaneously in the same amount of time. This approach reduces power since the power required for settling analog signal scales quadratically with the inverse of settling time. In our approach we assume that there are N/2 wavelet compression sub-modules located under the imager array. Each of these sub-modules reads in individual 2 x 2 matrices taken from the pixels, stores this data, and applies the appropriate lowpass and highpass filter operations. With this approach the downsampling and convolution (see Figure 2) are performed simultaneously. Thus the storage requirements for the convolution of the filters and the image is reduced from length1 + length2 - 1 elements to simply four results. Length1 and length2 are variables representing the number of elements in the two vectors to be convoluted. Although the wavelet transform is expressed in terms of lowpass and highpass filters, it could as easily have been presented in terms of averaging and differencing, which is the cornerstone of this hardware implementation. Thus, the operations previously implemented using software should readily be attainable using hardware. We will focus mainly on the lowpass lowpass subband of a 2 x 2-pixel array in our initial phase of producing the chip (see Figure 3a). However, by repeating the lowpass lowpass module, all four subbands can easily be produced for an N x N image, where N is the desired dimension of the picture to be compressed (see Figure 3b). It is evident from Figure 3b that the only difference between the Lowpass Lowpass module and the other subband modules is that the "r" (reference level) and "s" (signal level) values are in different locations, and placing the "r" and "s" values in different locations simply allow the difference coefficients to be produced. The left and right hand sides are composed of an array of "r" and "s" values, respectively in Figure 3a. The parameter "r" represents the reference level of the pixel or the initial charge stored in the pixel while s corresponds to the pixel signal plus the reference level. To get the value of the true signal taken from the pixel array, the difference "r-s" must be performed. By convention, in Figure 3a or Figure 3b the bus on the left-hand side is considered positive, while the bus on the right hand side is considered negative. This convention will be important when the values are routed off-chip to the differencing operational amplifier. In addition, Figure 3b contains calculations that the differencing operational amplifier will be performing to produce values for Lowpass Lowpass, Highpass Highpass and the rest of the Subband Coding modules. The capacitors are responsible for performing all on-chip arithmetic operations. Each of the capacitors in Figures 3a and 3b will be assigned the same capacitance to add proper weighting to the wavelet coefficients, as there exists a direct relationship between the capacitance and wavelet coefficients. Recall that in the 2-coefficient Haar, each coefficient has the same magnitude, thus each capacitor should have the same capacitance. In Figure 3a, the average of each array can be obtained by placing the four capacitors in parallel. The following relation between, q, c, and V illustrate this point. In equation (1), we formulate the general relation between q, c, and V

$$q \equiv cV\,, \tag{1}$$

in which the parameters q, c, and V correspond to the charge, capacitance and voltage, respectively. If two capacitors with the same capacitance C are connected in parallel, the result is

$$Q1 + Q2 \equiv 2C\overline{V}\,. \tag{2}$$

Equation (2) reflects charge conservation and capacitance doubling. As a result, the average voltage of two capacitors can be represented by

$$\overline{V} \equiv \left(Q1 + Q2\right) / \, 2C\,. \tag{3}$$

If these averages are routed onto bus lines and then off chip to a differencing operational amplifier, then the necessary addition and subtraction can be done. An example of this case is shown in Figure 3a. In theory other filters can be realized by using more capacitors and by scaling the capacitors, but both require more area. From a hardware standpoint, the 2-coefficient Haar wavelet lends itself to an area efficient implementation, because of its small number of taps, and also because the taps themselves are equal in size. To achieve a filter where one coefficient is ten times larger than another requires that one capacitor be ten times larger than another. When one considers the precision of the coefficients necessary for many image compression filters, the area required to realize them becomes prohibitive.

RESULTS AND DISCUSSION

The circuit in Figure 4 simulates the Lowpass Lowpass module mentioned in the above section. Basically the circuit takes voltages from the pixel array and averages them. The pixel arrays are simulated using voltage sources connected to switches that allow the pixel arrays to simply charge the capacitors. Each transistor has a gate voltage that marks the highest value that the circuit can average correctly. SPICE data was taken using the circuit in Figure 4 and the test image in matrix 1.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 2 & 2 & 2 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(Matrix 1)

Matrix 1, which corresponds to the test image, generates the image in Figure 5. The test matrix is fed into the software algorithm and the result is the production of four subbands, among which is the lowpass lowpass subband, i.e., the average coefficients as shown below in Matrix 2, which contains the theoretical values.

$$\begin{bmatrix} 0.0 & 1.0 & 1.0 & 0.0 \\ 0.0 & 2.0 & 2.0 & 0.0 \\ 1.0 & 3.0 & 2.0 & 0.0 \\ 1.0 & 2.0 & 1.0 & 0.0 \end{bmatrix}$$

(Matrix 2)

The same test matrix is fed into the circuit in Figure 4, yielding the results in Matrix 3, which contains the experimental values.

$$\begin{bmatrix} 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & 1.0 & 1.0 & 0.0 \\ 0.5 & 1.5 & 1.0 & 0.0 \\ 0.5 & 1.0 & 0.5 & 0.0 \end{bmatrix}$$

(Matrix 3)

It is noted that these values are exactly half of the software values due to the circuit scaling of ¼ and software scaling of ½ Therefore, the two matrices only differ by a constant, which is exactly the predicted outcome. Thus, the theoretical and experimental values agree within a constant.

CONCLUSION

We have presented a low power, area efficient implementation of the two taps Haar wavelet compression algorithm. Our results show that the results obtained with software simulations can effectively be implemented in hardware. The theoretical values produced from the software implementation differ only by a constant from those taken from the circuit simulation. Results suggest that an algorithm capable of 8:1, 4:1, and 2:1 compression with error measurements of 37 dB, 37.3 dB, and 39 dB, respectively can effectively be used to develop a functional hardware prototype. Although the hardware limitations restrict the software implementation; the results compare favorably to those of other unrestricted software algorithms. The circuit was simulated using MicroSim, and the layout is being implemented using Tanner Tools Pro Suite. Currently, the chip is in the secondary design phase. Future efforts will be directed towards completing the layout of the chip and submitting it to MOSIS for fabrication.

ACKNOWLEDGMENTS

Author, J.S.H. would like to thank program manager, Dr. Robert Ferraro, Manager, NASA Remote Exploration and Experimentation Project and Associate Manager, NASA Earth and Space Sciences Project for support and efforts. In addition, authors L.A.W, J.S.H and K.R.P wish to thank the Jet Propulsion Laboratory, Center for Space Microelectronic Technology and Dr. Roman Murenzi, Associate Professor, Clark Atlanta University, Department of Physics for their contributions at various stages of the project.

REFERENCES

[1] B. Pain, R.H. Nixon, S.E. Kemeny, C.O. Staller and E.R. Fossum, "256x256 CMOS Active Pixel Sensor Camera-On-A-Chip", Center for Space Microelectronics Technology, JPL-California Institute of Technology, JPL Technical Report.

[2] Lanier Watkins, *Modulation Characterization Using the Wavelet Transform*, Master Thesis, Clark Atlanta University, May 1997.

[3] R. Murenzi, K. Bouyoucef, J.P. Antoine and P. Vandergheynst, "Alternative Representations of an Image via the 2-D Wavelet Transform. Application to Character Recognition", SPIE, Vol. 2488, April 17-18, 1995, Orlando, Florida.

[4] K.R. Namuduri, S.G. Romaniuk and N. Ranganathan, "A Lossless Image Compression Algorithm using Variable Block Size Segmentation", IEEE Trans. On Image Processing, Vol. 4, No. 10, 1396-1406, October 1995.

[5] G. Strang and T. Nguyen, Wavelets and Filter Banks, Prentice Hall, 1992.

[6] Jerome Shapiro "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", IEEE Transactions on Signal Processing, Vol. 41, No. 12, Dec. 1993.

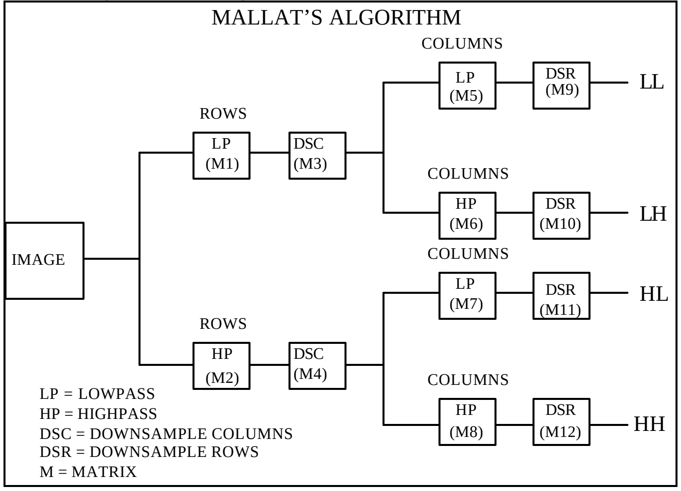

Figure 1. Mallat's Algorithm

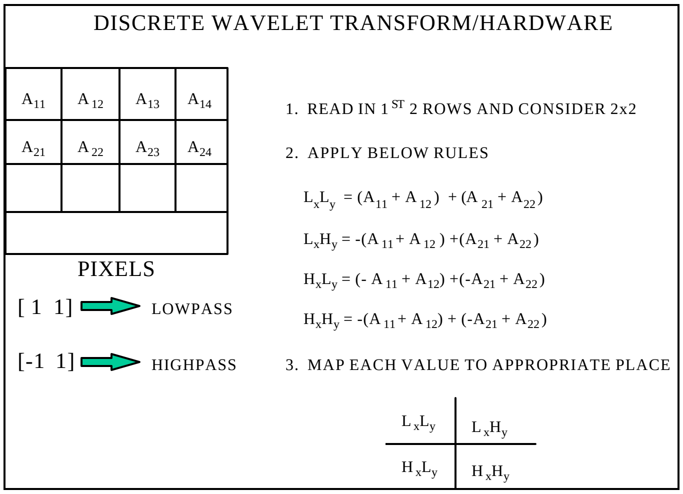
DISCRETE WAVELET TRANSFORM/HARDWARE
A_{11} A_{12} A_{13} A_{14}
A_{21} A_{22} A_{23} A_{24}
PIXELS
[1 1] LOWPASS
[-1 1] HIGHPASS
1. READ IN 1 ST 2 ROWS AND CONSIDER 2x2
2. APPLY BELOW RULES
$L_xL_y = (A_{11} + A_{12}) + (A_{21} + A_{22})$
$L_xH_y = -(A_{11} + A_{12}) + (A_{21} + A_{22})$
$H_xL_y = (-A_{11} + A_{12}) + (-A_{21} + A_{22})$
$H_xH_y = -(A_{11} + A_{12}) + (-A_{21} + A_{22})$
3. MAP EACH VALUE TO APPROPRIATE PLACE
L_xL_y L_xH_y
H_xL_y H_xH_y

Figure 2. Discrete Wavelet Transform/Hardware

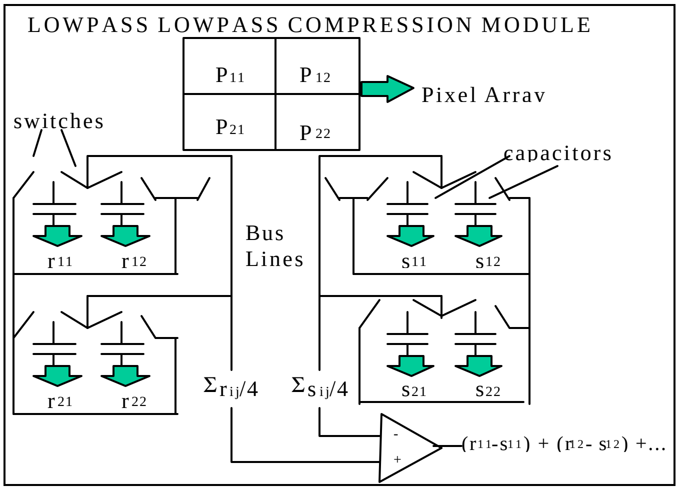

Figure 3a. Lowpass Lowpass Compression Module

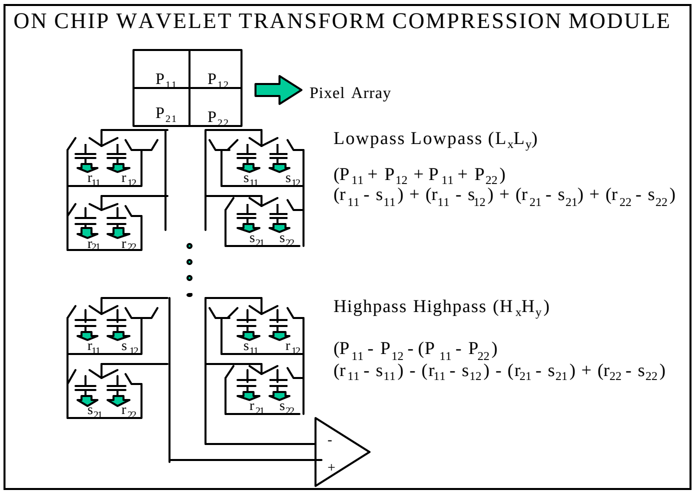
ON CHIP WAVELET TRANSFORM COMPRESSION MODULE
P_{11} P_{12}
P_{21} P_{22}
Pixel Array
r_{11} r_{12} s_{11} s_{12}
r_{21} r_{22} s_{21} s_{22}
Lowpass Lowpass (L_xL_y)
$(P_{11} + P_{12} + P_{11} + P_{22})$
$(r_{11} - s_{11}) + (r_{11} - s_{12}) + (r_{21} - s_{21}) + (r_{22} - s_{22})$
r_{11} s_{12} s_{11} r_{12}
s_{21} r_{22} r_{21} s_{22}
Highpass Highpass (H_xH_y)
$(P_{11} - P_{12} - (P_{11} - P_{22})$
$(r_{11} - s_{11}) - (r_{11} - s_{12}) - (r_{21} - s_{21}) + (r_{22} - s_{22})$
-
+

Figure 3b. Wavelet Transform Compression Module

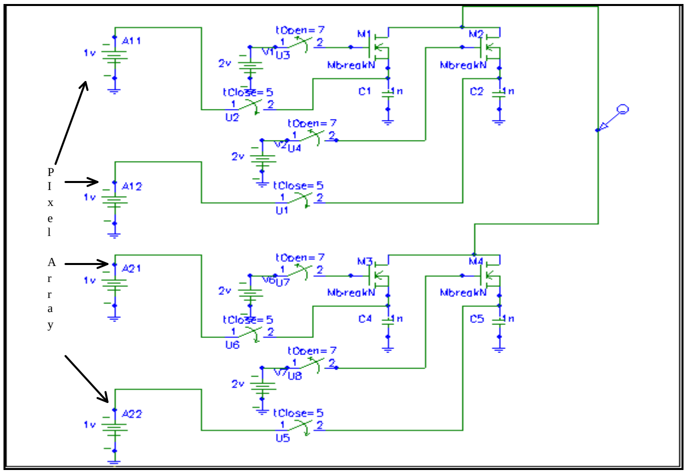

Figure 4. Image Compression MicroSims Schematic

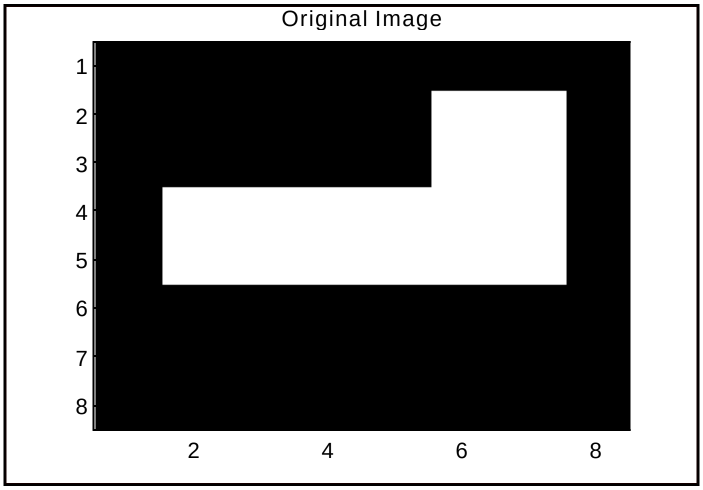

Figure 5. Test Image