ADVANCES IN IMAGE AND VIDEO PROCESSING
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 3, ISSUE 1
ISSN: 2054-7412

An Algorithm with Low Complexity for Image Compression and its Hardware Implementation using VHDL

Shaktijeet Mahapatra[1] and Mihir Narayan Mohanty[2]
Department of ECE, Institute of Technical Education and Research (ITER), Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, India
[1]shaktijeetmahapatra@soauniversity.ac.in; [2]mihirmohanty@soauniversity.ac.in

ABSTRACT

Image compression is highly essential for efficient transmission and storage of images in the field of communication engineering, bio-medical applications. Also, the compression technology is of special interest for the fast transmission and real-time processing on the internet. For reduced form and less capacity, the area of research growing day by day. The objective of image compression is to find a new representation in which pixels are less correlated, but with the original contents. In this paper, the existing as well as new algorithms are applied for compression for evaluation. The results have been compared for both techniques. On the basis of evaluating and analyzing the image compression techniques it presents the VHDL implementation of low complexity 2D-DWT approach applied to image compression. The decompression has to invert the transformations applied by the compression to the image data. When using the wavelet transform it is possible to exploit the unique properties of the wavelet coefficients to efficiently encode them.

Keywords. Image compression, real-time processing, VHDL implementation, 2D-DWT.

1 Introduction

Images constitute the significant part of data, particularly in remote sensing, biomedical and video conferencing applications. Image compression is the reduction or elimination of redundancy in data representation in order to achieve reduction in storage and communication cost. Image compression techniques reduce the number of bits required to represent an image by taking advantage of these redundancies. Image compression addresses the problem of reducing the amount of data required to represent a digital image. It is a process intended to yield a compact representation of an image, thereby reducing the image storage/transmission requirements.

As the bandwidth requirement increase, the transmission or storage cost also increase simultaneously, so it is necessary to employ compression techniques, which reduce the data rate while maintaining the subjective quality of the decoded image or video signal. Even with the advances in bandwidth and storage capabilities, if images were not compressed many applications would be too costly. With the use of digital cameras, requirements for storage, manipulation, and transfer of digital images, has grown exponentially. These image files can be very large and can occupy a lot of memory.

It provides a potential cost savings associated with sending less data over switched telephone network where cost of call is really usually based upon its duration.

• It not only reduces storage requirements but also overall execution time.

DOI: 10.14738/aivp.31.800
Publication Date: 09th January, 2015
URL: http://dx.doi.org/10.14738/aivp.31.800

• It also reduces the probability of transmission errors since fewer bits are transferred.

• It also provides a level of security against illicit monitoring.

2 Related Literature

A DCT-based method is specified for "lossy" compression, and a predictive method for "lossless" compression. JPEG features a simple lossy technique known as the Baseline method, a subset of the other DCT-based modes of operation. The Baseline method has been by far the most widely implemented JPEG method to date, and is sufficient in its own right for a large number of applications. [1]. In [2] for analyzing image compression methods that are based on wavelet decompositions. The theory relates the rate of decay in error between original image and the compressed images that is measured using a family of L^p norms, as the size of compressed image increases, to the smoothness of the image (in certain smoothness classes called Besov's spaces). Within this theory, error incurred by the quantization wavelet transform coefficients is explained. An adaptive lossy LZW algorithm is proposed for palettised image compression in [3]. The algorithm employs an adaptive thresholding mechanism with human visual characteristics to lessen distortion. A probability model for natural images based on empirical observation of their statistics in wavelet domain has been developed and presented in [4]. Similarly some algorithms have been proposed by researchers for lossy and nearly lossy compression techniques [5-6]. An efficient VLSI design of one-dimensional direct discrete wavelet transform processor is presented in [7]. A real-time wavelet image compression algorithm using vector quantization and its VLSI architecture are proposed in [8]. The proposed zero-tree wavelet vector quantization (WVQ) algorithm focuses on the problem of how to reduce the encode wavelet images with high coding efficiency. Lossless compression is usually required in the medical image field is presented in [9]. The word length required for lossless compression makes too expensive the area cost of the architectures that appear in the literature. SPIHT algorithm is used to develop the codec, where authors claimed for their better efficiency [10]. The embedded block coding algorithm at the heart of the JPEG 2000 image compression standard is described in [11]. A lossless compression of images using coding schemes and patterns that include minterm, cube and coordinate data coding, Walsh, triangular and Reed–Muller weights based patterns, Reed–Muller spectra and reference row technique is proposed in [12]. A lossless wavelet-based image compression method with adaptive prediction is proposed in [13]. Also, wavelet transform is used with neural network by authors in past [14]. Signal processing methods have been used in FPGA implementation using VHDL by many authors [15-17]. But a considerable work on image processing methods has been implemented in FPGA [18-19]. An FPGA-based image and data processing core for future generation wireless capsule endoscopy (WCE) is presented in [20]. A resource efficient and high-performance architecture for a two-dimensional multi-level discrete wavelet transform processor is presented in [21]. Therefore authors were motivated to work in this field and hence taken the choice with wavelet transform.

The most commonly used metrics still remain simple, mathematically defined measures such as peak signal to noise ratio (PSNR) or mean squared error (MSE). When the quantization is varied on a single image in a straightforward manner, such as by varying the scale factor in JPEG compression, these metrics do correlate with image quality.

3 Method for Compression

The wavelet transform has been widely used in image and video compression since it allows localization in both the space and frequency domains [22]. The intensive computation of DWT due to its inherent multilevel data decomposition and reconstruction operations brings a bottleneck that drastically reduces its performance and implementations for real-time applications when facing large size digital images and/or high-definition videos. A hardware implementation of 1-D NHWT was presented in [23]. The architecture described in [20] involves only additions and subtractions along with normalization but only along one dimension.

A hardware implementation of 2-D Haar wavelet transform has been attempted in this work. The computing strategy finally chosen for coding the 2-D Haar wavelet transform is based on first evaluating the one-dimensional Haar wavelet transform to each row of pixels (a matrix of intermediate results is thus obtained), followed by the evaluation of the one-dimensional transform of the columns of the resulting intermediate matrix.

Two main aims have been pursued in the conception of this implementation: first, to exploit as much as possible the local properties of the Haar wavelet in order to eliminate possible redundant information during the coding process; and, second, to make use of the inherent arithmetical simplicity of the Haar wavelet transform in order to attain a low circuit complexity hardware implementation. The main aim is to implement a complete processor using one configurable general-purpose chip, yielding the lowest possible loss of quality. The simulation results have been presented thereafter. The VHDL implementation was done using Xilinx ISE 14.1.

3.1 Implementation of JPEG algorithm

The JPEG compression and decompression algorithm was coded without using any MATLAB library functions so that a hardware implementation using VHDL could be developed that conforms to the algorithm. The algorithm was implemented in following phases:

1) The uncompressed source data is separated into 8 × 8 blocks of pixels. 128 is subtracted from the value of each pixel so that the new effective range is from -128 to 127.
2) Each block is transformed into an 8 × 8 block of frequency coefficients using DCT.
3) These coefficients are quantized, by dividing the frequency coefficient matrix by the quantization matrix on an element-by-element basis.
4) An entropy encoder is applied to the quantized coefficients. The algorithm uses a zigzag ordered encoding, which collects the high frequency quantized values into long strings of zeros. DC values use delta encoding, which means that each DC value is compared to the previous value, in zigzag order. For all other values, i.e. high frequency values, Huffman coding (or any run-length coding is used).

The comparison between original and decompressed image has been based on parameters like mean square error (MSE), root mean square error (RMSE) and peak signal to noise ratio (PSNR) values.

3.2 Implementation of DWT (Haar Transform)

In this work, Haar wavelets compression is used. It provides for an efficient way to perform both lossless and lossy image compression. It relies on averaging and differencing values in an image matrix which may be sparse or nearly sparse. A sparse matrix has large number of entries as zeros, which can be stored in an efficient manner, leading to smaller file sizes. The algorithm to compress the image is as follows:

1. An image is sub-divided into blocks of 8x8 pixels (with padding if necessary).
2. In each of the blocks, the pixels are grouped in twos. Then first 4 columns are replaced by their averages and last four by the half of their differences. The first four coefficients are known as approximation coefficients and last four are called detail coefficients.
3. In the next step, the first entries are again grouped in twos and first two are replaced by the averages and next two by the half of the differences, leaving last 4 entries unchanged.
4. In this step, the first 2 entries are grouped and replaced by their average and half of their difference.
5. The steps are repeated for the remaining rows of the block. After this, the process is repeated for the columns, grouping rows in the same manner as columns.

An easier implementation of 2D Haar Transform involves two 8x8 matrix multiplications, i.e., H^TAH, per 8x8 pixel block of the image. But from a hardware point of view, this procedure requires a total of 1024 multiplications and 896 additions and a subsequent division by 8, 4, and2 operations for generating a resulting 8x8 matrix. For a lossless implementation, this also involves floating point implementation of this transform.

The implementation presented in this work, is a multiplier-less implementation of a lossy 2D Haar transform and involves only 112 additions, 112 subtractions and 224 shift-right-by-1-bit operations on 8-bit signed numbers. The addition and subtraction of the numbers have been implemented in the same module, thereby reducing the number of components having different design. The subtraction of two numbers and the addition of the negative numbers have been done using 2's complement of the binary numbers using 8-bit binary adder/ subtractor. The operations involving divide-by-2 has been achieved using shift-right-by-1-bit operation after each addition or subtraction. This solves two major purposes: first, this eliminates the need for floating point implementation; and, second, it drastically reduces computation load on the module. The data considered here, is a block of 8x8 pixels, which are of 8 bits each, corresponding to gray-scale intensity levels of 0-255 in case of unsigned numbers, or -128 – 127 in case of signed numbers.

The compression and decompression of the images using DWT was coded without using any MATLAB library functions so as to aid coding for hardware implementation using VHDL.

4 Simulation Results

From Table 1, it can be seen that JPEG algorithm takes considerable time for execution. This can be attributed to the fact that in phase 2, DCT is applied to the pixel blocks which is one of the most computation intensive phases; and in phase 4, wherein Huffman coding is applied which again is a computation intensive step. Same can be said for decompression times. Images with higher redundancy are compressed to a greater extent as compared to images with lower redundancy. Thus the compression ratio varies for different images. It can be seen that lossy JPEG compression produces error, but the error occurs mostly in high frequency elements.

As can be seen from Table 1, run times for compression and decompression using LZW algorithm are somewhat lesser than that of JPEG algorithm. This is because of the fact that, in LZW algorithm only encoding or decoding of the sequence compresses or decompresses the sequences without applying any transforms, which saves some time.

Table 1: Observed Average Run-Time for JPEG Baseline, LZW and DWT (Haar) algorithms

Image	Average Run Time for Compression (in seconds)			Average Run Time for Decompression (in seconds)		
	JPEG Baseline Algorithm	LZW Baseline Algorithm	DWT Algorithm	JPEG Baseline Algorithm	LZW Baseline Algorithm	DWT Algorithm
Cameraman.tif	39.84	23.16	0.094	70.56	13.20	0.234
Eye_gray.bmp	41.96	15.03	0.091	42.83	10.76	0.228
Woman_blonde.tif	52.84	15.14	0.093	74.62	14.04	0.23
Woman_darkhair.tif	35.53	15.18	0.090	61.41	9.88	0.229
Pirate.tif	38.20	16.10	0.098	67.59	11.77	0.228
Mandril_gray.tif	40.00	21.54	0.095	68.95	21.56	0.227

The images that have been used are 512x512 pixel size gray scale, standard test images. In Figure 1 and Figure 2, the left picture is original picture and the right one is the reconstructed picture after compression and decompression.

1. The Cameraman picture

Figure: 1: Grayscale 512x512 Cameraman image [24] (a) Original image; Reconstructed image after lossy compression and decompression using DWT (b) with ε = 2, (c) with ε = 5, (d) with ε = 10.

2. Other test pictures

Figure: 2: (1) Grayscale 512x512 Eye gray image, (2) Grayscale 512x512 Woman blonde image, (3) Grayscale 512x512 Woman_darkhair image, (4) Grayscale 512x512 Pirate image, (5) Grayscale 512x512 Mandril_gray image [24]

It can be observed that the run time for compression and decompression for DWT algorithm are less than even a second. The observed run-time was equal both for lossless and lossy cases. This can be attributed to the fact that DWT using Haar transforms do not use any transforms like DCT or any encoding technique. This makes it a choice for hardware implementation in this work.

Table 2: Observed compression ratios and error in case of JPEG

Image	Compression Ratio	MSE	RMSE	PSNR (in dB)
Cameraman.tif	1.84	34.23	5.85	75.57
Eye_gray.bmp	2.18	27.50	5.24	77.76
Woman_blonde.tif	1.55	162.94	12.76	59.97
Woman_darkhair.tif	2.60	31.08	5.57	76.53
Pirate.tif	1.58	44.04	6.63	73.05
Mandril_gray.tif	1.32	118.34	10.88	63.16

Table 3: Compression Ratio and Errors observed in case of DWT (lossy case, ε = 2)

Image	Compression Ratio	MSE	RMSE	PSNR (in dB)
Cameraman.tif	6.20	4.99	2.23	94.82
Eye_gray.bmp	6.78	8.30	2.88	89.73
Woman_blonde.tif	3.22	8.79	2.96	89.17
Woman_darkhair.tif	8.91	6.84	2.61	91.66
Pirate.tif	3.16	8.05	2.83	90.04
Mandril_gray.tif	2.09	7.37	2.71	90.93

Table 4: Compression Ratio and Errors observed in case of DWT (lossy case, ε = 5)

Image	Compression Ratio	MSE	RMSE	PSNR (in dB)
Cameraman.tif	14.45	22.67	4.76	79.69
Eye_gray.bmp	22.12	28.83	5.37	77.28
Woman_blonde.tif	9.19	39.28	6.27	74.19
Woman_darkhair.tif	29.29	22.21	4.71	79.89
Pirate.tif	8.69	43.49	6.59	73.17
Mandril_gray.tif	4.88	56.07	7.49	70.64

Table 5: Compression Ratio and Errors observed in case of DWT (lossy case, ε = 10)

Image	Compression Ratio	MSE	RMSE	PSNR (in dB)
Cameraman.tif	29.35	64.43	8.02	69.25
Eye_gray.bmp	46.81	54.27	7.37	70.96
Woman_blonde.tif	23.06	97.80	9.88	65.07
Woman_darkhair.tif	49.56	45.28	6.73	72.77
Pirate.tif	23.66	113.96	10.67	63.54
Mandril_gray.tif	15.29	190.65	13.80	58.39

In Table 2, it can be observed that the run time for compression and decompression are the less than even a second. The observed run-time was equal both for lossless and lossy cases. This can be attributed to the fact that DWT using Haar transforms do not use any transforms like DCT or any encoding technique. This makes it a choice for hardware implementation in this work.

In Tables 3 through 5, it can be clearly seen that compression ratio achieved using lossy compression techniques is greater than that achieved using lossless technique. The compression ratio can further be increased by making more number of entries to be zero. A non-negative threshold, ε can be fixed and any entry in the compressed matrix which is less than ε can be made zero. Hence resulting in a more sparse matrix and hence more saving in space. Compression ratios of about 20:1 are easily achievable. Higher compression ratios come at the cost of picture quality. It can be seen that the mean square error (MSE) is highest when ε = 15, thus picture quality degrades. The lossless case, by nature, will yield zero MSE.

4.1 VHDL Implementation

Figure 3 shows the schematic of how Haar transform has been applied in order to achieve the objective.

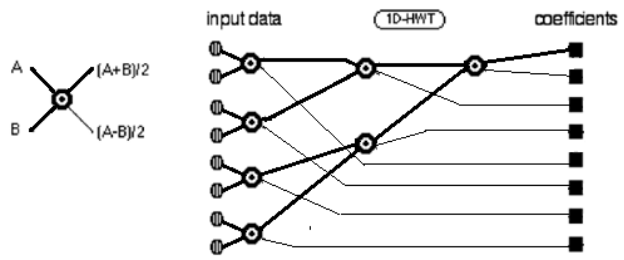

Figure 3: Computation scheme for discrete 1-D Haar wavelet transform

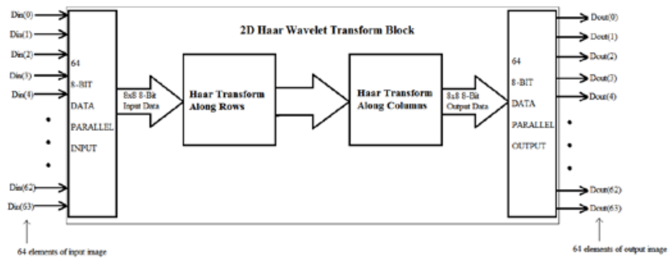

Figure 4: Schematic of the hardware implementation of 2D Haar Transform.

The data is brought into the module by 64 8-bit lines with a view to reduce the delay involved in fetching the data elements into the module.

Simulation Results

An 8 x 8 image has been used for the purpose of simulation.

The advanced HDL synthesis report as generated by Xilinx shows that 224 adder/subtractor blocks were totally needed.

```
=========================================================================
*                        Advanced HDL Synthesis                         *
=========================================================================


=========================================================================
Advanced HDL Synthesis Report

Macro Statistics
# Adders/Subtractors                                   : 224
 8-bit addsub                                          : 224
# Registers                                            : 512
 Flip-Flops                                            : 512

=========================================================================
```

Figure 5: Advanced HDL Synthesis report generated by Xilinx ISE

A 20 ns simulation shows that the results have been somewhat satisfactory. This simulation shows that there are many non-zero coefficients occurring in the matrix. This condition stems from the fact that no threshold co-efficient has been applied. No attempt has been made to make the detail coefficients to be zero.

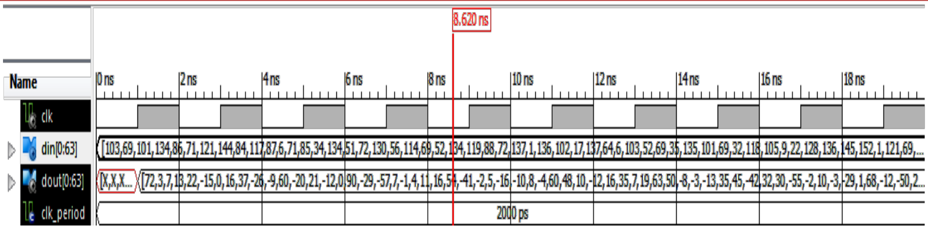

Figure 6: Test bench Simulation results

5 Conclusion

The DWT technique of compression has also been applied to the images. By the choice of Haar wavelet, it was found that there was minimal amount of error involved in case of lossy compression, which is responsible for higher PSNR values. A multiplier-less 2D Haar wavelet transform implementation was presented, which turns out to be very promising in reducing system complexity in terms of time and area, given the fact that DWT is computationally very costly. This can be taken care in future work by developing even simple algorithm.

REFERENCES

[1] G. K. Wallace, "The JPEG Still Picture Compression Standard", Vol. 38, no. 1, IEEE Transactions on Consumer Electronics, pp. xviii – xxxiv, Feb. 1992.

[2] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445-3462, December 1993.

[3] S.W. Chiang, and L.M. Po, "Adaptive Lossy LZW Algorithm for Palettised Image Compression", Electronics Letters, IEE, pp. 852-854, March 1997.

[4] R. W. Buccigrossi, E.P. Simoncelli, "Image Compression via Joint Statistical Characterization in Wavelet Domain", Vol. 8, No. 12, pp. 1688-1701, December 1999.

[5] M. Boliek, M. J. Gormish, E. L. Schwartz and A. Keith, "Next generation image compression and manipulation using CREW," *Proceedings of the IEEE International Conference on Image Processing,* October 26-29, Santa Barbara, CA, pp. III-567-III-357, 1997.

[6] M. J. Gormish, E. L. Schwartz, A. Keith, M. Boliek and A. Zandi, "Lossless and nearly lossless compression for high quality images," *Proceedings of the SPIE/IS&T Conference on Very High Resolution and Imaging II*, vol. 3025, San Jose, CA, pp. 62-70, February 1997.

[7] C. Souani, M. Abid, K. Torki, R. Tourki, "VLSI design of 1-D DWT architecture with parallel filters", INTEGRATION, the VLSI Journal 29, pp. 181-207,2000.

[8] S-K Paek, L-S Kim, "A Real-Time Wavelet Vector Quantization Algorithm and Its VLSI Architecture", pp. 475-489, Vol. 10, No. 3, IEEE Transactions on Circuits and Systems for Video Technology, April 2000.

[9] Urriza, et al., "VLSI Implementation of Discrete Wavelet Transform for Lossless Compression of Medical Images", Real-Time Imaging 7, pp. 203-217, 2001.

[10] A. Said and W. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243-50, June 1996.

[11] D. Taubmann, E. Ordentlich, M. Weinberger, G. Seroussi, "Embedded Block Coding In JPEG2000", Signal Processing: Image Communication 17, pp. 49-72, 2002.

[12] B. J. Falkowski, "Lossless Binary Image Compression Using Logic Functions and Spectra", Computers and Electrical Engineering 30, pp. 17-43, 2004.

[13] Y-T Chen, D-C Tseng, "Wavelet Based Medical Image Compression with Adaptive Prediction", Computerized Medical Imaging and Graphics 31, pp. 1-8, 2007.

[14] Rasmita lenka, Swagatika Padhi, Minakshee Behera, Naresh Patnaik, Mihir N. Mohanty, "Design of Neuro-wavelet based vector quantizer for image compression", Special Issue of International Journal of Computer and Communication Technology, Volume 1 Issue 2, 3, 4; pp- 214-221,August-2010.

[15] Asit Kumar Subudhi, Biswajit Mishra, Mihir N. Mohanty**,** "VLSI Design and Implementation for Adaptive Filter using LMS Algorithm", IJCCT, Vol-2, Issue-6, 23rd -24th Feb, 2011.

[16] Mihir Narayan Mohanty, Biswajit Mishra, Aurobinda Routray, "FPGA implementation of CLMS Algorithm", ICEAS, IEEE Conference, Bhubaneswar, 28th -30th December, 2011.

[17] Mihir N. Mohanty, Hemanta Kumar Sahu, "FPGA Implementation of Variable Step-size LMS Algorithm", IEEE Conf.- ICCSP' 13**,** Melmaruvathur, TN, 03- 05 April 2013.

[18] Panchami Padmasana, Mihir Narayan Mohanty, Hemanta Kumar Sahu, "VHDL Implementation of Spatial Filter for Image Enhancement", IEEE Conf., Chennai, ICCSP – 3-5 April 2014.

[19] Panchami Padmasana, Mihir Narayan Mohanty, P. Kabisatpathy, "FPGA Implementation of Modified Median Filter for Impulse Noise Removal from Image", IJECT Vol. 5.3 - Spl 1, July - September, 2014.

[20] P. Turzca, M. Duplaga, "Low Power FPGA-Based Image Processing Core for Wireless Capsule Endoscopy", Sensors and Actuators A: Physical 172, pp. 552-560, 2011.

[21] S.M. Aziz, D.M. Pham, "Efficient Parallel Architecture for Multi-Level Forward Discrete Wavelet Transform Processors", Computers and Electrical Engineering 38, pp. 1325-1335, 2012.

[22] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine intelligence,* vol. 11, no. 7, pp. 674-693, July 1989.

[23] F. Javier Diaz, A. M. Buron, J. M. Solana, "Haar wavelet based processor scheme for image coding with low circuit complexity", Computers and Electrical Engineering 33, pp. 109-126, 2007.

[24] http://www.imageprocessingplace.com/root_files_V3/image_databases.htm