Published in Proceedings of the "IEEE Nordic Signal Processing Symposium NORSIG 2000", 17-20, 2000
which should be used for any reference to this work

A MULTIPLIERLESS IMPLEMENTATION SCHEME FOR THE JPEG IMAGE CODING ALGORITHM

Javier Bracamonte, Patrick Stadelmann, Michael Ansorge, and Fausto Pellandini

Institute of Microtechnology, University of Neuchâtel
Rue A.-L. Breguet 2, 2000 Neuchâtel, Switzerland
Phone: +41 32 7183426; Fax: +41 32 7183402
Email: javier.bracamonte@imt.unine.ch

ABSTRACT

This paper reports an efficient implementation scheme for a JPEG encoder. The baseline JPEG algorithm is executed without involving any multiplication. All the arithmetic operations are reduced to simple additions/subtractions and very short shifts. This translates into a hardware implementation of reduced complexity, which makes this approach attractive in digital image applications for portable devices, where silicon area and power consumption are dominant issues in the design. Furthermore, this multiplierless implementation produces negligible losses in terms of compression efficiency, as well as in terms of objective/subjective quality of reconstructed images, with respect to a JPEG system that executes standard integer multiplications.

1. INTRODUCTION

In search of reducing the implementation complexity of algorithms, multiple approaches have been proposed in the signal processing literature. Given that the multiplications are usually the more expensive operations from a hardware implementation point of view, it is worth dedicating some effort to circumvent their direct execution and replace them by simpler operations.

The multiplierless approach has been successfully applied in the design of digital filters or in coding systems that rely on structures whose basic element is an FIR filter with a reduced number of taps [1,2,3]. In this paper the no-multiplication approach is studied for a system of a relatively higher complexity, as is the case of an industry standard image encoder. The results to be presented show that it is possible to implement an elaborate image compression system with a reduced amount of hardware resources.

This paper is organized as follows. Section 2 briefly describes the JPEG algorithm. Section 3 reports the multiplierless approach, including the discussion of some design choices. Section 4 elaborates on the multiplierless approximation for the different modules of the JPEG encoder. Section 5 illustrates an implementation strategy that will reduce the maximum individual shift required by the add/shift operator, which means a simpler shifter hardware module. Different approximation schemes are reported in Section 6, along with their results and comparisons with those produced by a 16-bit standard-integer-multiplication implementation (SIMI). Finally, the conclusions are stated in Section 7.

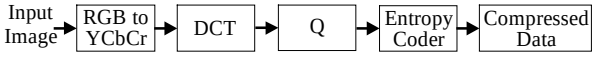


Figure 1. Baseline JPEG encoder

2. THE JPEG ALGORITHM

Figure 1 shows the block diagram of the sequential or baseline JPEG encoder. The following paragraphs describe the main operations of the JPEG algorithm [4].

The first operation is related to a color space transformation. The input RGB image is converted into the luminance/chrominance YCbCr color space representation. This transformation favors compression efficiency since a significant part of the interband RGB correlation is eliminated.

The luminance and chrominance bands are then subdivided into blocks of 8x8 pixels. On each of these blocks, a two dimensional 8-point Discrete Cosine Transform (DCT) is applied in order to decorrelate the original data.

After the linear transform operation, the resulting 2-D DCT coefficients are quantized with a 64-element normalization matrix. The visually important 2-D DCT coefficients, located in the top left region of the array, are quantized with short quantization steps while the rest of the coefficients is coarsely quantized.

An entropy coding stage follows the quantization unit; for baseline JPEG, the entropy coder is basically executed with a lossless differential PCM operation and a particular implementation of a Huffman encoder. Since the entropy coder does not involve any multiplication, its implementation will not be discussed in the remaining of this paper.

3. MULTIPLIERLESS APPROACH

The multiplierless approach starts by identifying all the multiplications, $y=ax$, required by the JPEG algorithm, where a represents a fixed coefficient, and x, a signal sample value. The set of a coefficients is then approximated and replaced by a corresponding set of $\hat{a}$ values.

Three constraints were defined to produce the set of $\hat{a}$

values. The first one was that each â should approximate its corresponding a coefficient with a maximum absolute relative error, |(a – â)/a|, of 1%. This accuracy requirement conditions the results of the multiplierless approach to be very close to those produced by a SIMI, in terms of compression ratio, as well as in terms of reconstruction quality.

The second constraint consisted in producing a binary representation of each â, which contains the least number of 1's. This implies the execution of y=âx with a minimum number of add/shift iterations or stages, which favors speed and power consumption.

In order to minimize the silicon area of the add/shifter unit, the maximum single shift required to represent each â should be kept to a minimum value. The maximum single shift was constrained in this implementation as not to be higher than 6. Such a short shift value is obtained by using the add/shift implementation described in Section 5.

4. MULTIPLIERLESS APPROXIMATION OF THE JPEG ENCODER

The following sections report the results of the multiplierless approximation for each of the different JPEG modules described in Section 2.

4.1. RGB to YCbCr conversion

Alike other image and video coding methods, JPEG compresses color images by converting the input RGB image into the YCbCr color space. This conversion is carried out by the following linear transformation:

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}$$

4.1.1. Multiplierless approximation

Table 1 shows the approximated values of the elements of the color space conversion matrix. The largest number of add/shift operations and the maximum individual shift is 4 in both cases. Using this value, a maximum relative error of 0.87% is obtained, which demonstrates that to obtain an excellent approximation on the color conversion a reduced amount of hardware resources is sufficient. The maximum required wordlength for the coefficients is 11 bits.

4.2. Discrete Cosine Transform

Given the property of separability of the 2-D DCT, in order to reduce the complexity of implementation, the computation of the bidimensional transform is normally separated into two consecutive 1-D DCT operations. First applied over the rows of the 8x8-pixel block, and then, over the columns of the just 1-D transformed array.

The 1-D DCT used in JPEG is defined in [4] as:

$$X_u = \frac{1}{2\alpha} \sum_{i=0}^{7} x_i \cos\left[(2i+1)u\pi/16\right]$$

for $u = 0,1,\ldots,7$, where $\alpha = \sqrt{2}$ for $u = 0$; otherwise $\alpha = 1$. X_u represents the value of the 1-D DCT coefficient at the point u in the transformed domain, and x_i represents the value of the input vector at the point i.

In order to speed up the computation of the DCT, several fast DCT algorithms have been developed. Among them, the algorithm proposed in [5] has become very popular in systems implementation because it requires a reduced number of multiplications. This fast DCT algorithm, whose flowgraph is shown in Figure 2 requires 5 multiplications and 29 additions.

As indicated by the scaling factors α and β at the output of the flowgraph in Figure 2, the DCT coefficients produced by this algorithm, are up-scaled values. The correction to non-scaled DCT coefficients is implicitly made by merging the corresponding down-scale values with the normalization coefficients of the quantization stage.

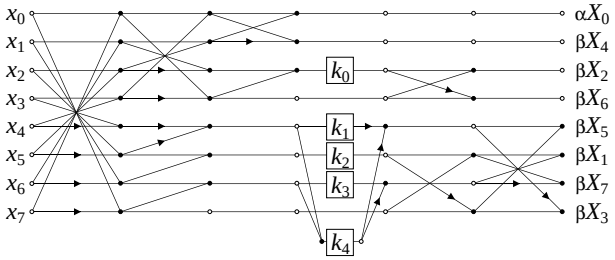


Figure 2. Fast Scaled Discrete Cosine Transform

4.2.1. Multiplierless approximation

Table 2 shows the approximated values of the multiplication coefficients of the fast scaled DCT algorithm (k_0=k_2 in Figure 2). The largest number of shifts is 4, and the maximum individual shift is equally 4. The maximum relative error produced is 0.56%, which again demonstrates that an excellent approximation can be obtained for the DCT operation with reduced hardware resources. The maximum required wordlength in this case is 9 bits.

For optimization purposes related to the implementation, the values corresponding to the second and third entries in Table 2 are one half of the corresponding values given in [5].

4.3. Quantizer

The quantization stage consists in dividing each 2-D DCT coefficient *Xi* by a normalization value *Qi* (or equivalently multiplying *Xi* by 1/*Qi*). A different quantization table is used for processing the luminance and the chrominance bands, which results in a total of 128 normalization values to be approximated. For the implementation reported in this paper the normalization tables recommended in [4], page 143, have been used. Besides, in order to obtain decompressed images of excellent quality, these coefficients have been multiplied by ½.

As mentioned in the previous section, the quantization values are also multiplied by the corresponding

down-scale values required to produce the correct non-scaled quantized DCT values.

4.3.1. Multiplierless approximation

Table 3 shows the statistics of the approximation of the 128 normalization coefficients. The maximum number of add/shift operations per coefficient is 6, which occurs in 5 quantizations. The overall maximum single shift required is also 6, which again occurs in 5 instances. The maximum relative error among the 128 approximation was found to be 0.98%. For this unit, the maximum required wordlength is 16 bits.

5. IMPLEMENTATION OF THE ADD/SHIFT OPERATION

The complexity of a shifter is proportional to the maximum shift that this module can execute. Thus, to reduce its complexity, the maximum individual shift required by the multiplierless system should be reduced to a minimum value. In order to achieve this goal, the canonic *sum of shifts* operation can be decomposed into a sequence of *accumulate-and-shift* operations, as illustrated in the following example.

The first coefficient in Table 1 is approximated with the binary number $\hat{a} = 0.0100110000_2$, which requires a maximum single shift of six to execute $y = \hat{a}x$:

$$y = \hat{a}x = x\cdot 2^{-2} + x\cdot 2^{-5} + x\cdot 2^{-6}$$

This maximum shift value can be reduced, if the inner minimum shifting value 2^{-min} is factored iteratively, which produces an accumulate-and-shift sequence of the following form:

$$y = \hat{a}x = 2^{-2}\cdot(x + x\cdot 2^{-3} + x\cdot 2^{-4})$$

$$y = \hat{a}x = 2^{-2}\cdot(x + 2^{-3}\cdot(x + x\cdot 2^{-1}))$$

Which reordered gives:

$$y = \hat{a}x = ((x\cdot 2^{-1} + x)\cdot 2^{-3} + x)\cdot 2^{-2}$$

This alternative implementation reduces the maximum required shift to 3: half the original value. The maximum shift figures given in Tables 1, 2, and 3 were calculated by using this add/shift implementation scheme.

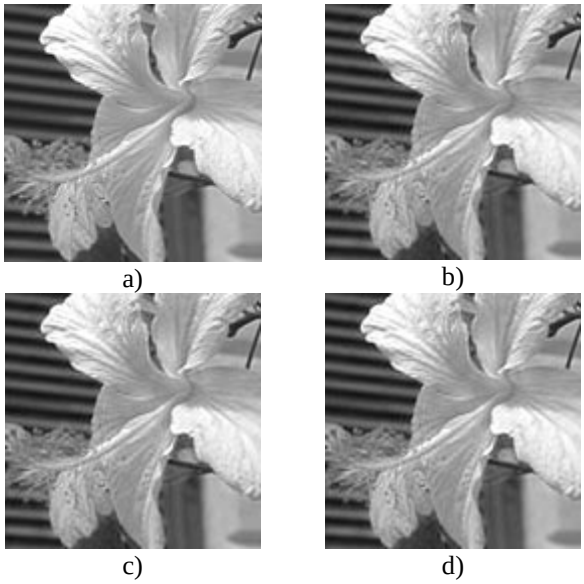
The shorter shifts required by the latter implementation, also make a more efficient use of the information bits located in the least significant part of registers, that would otherwise be lost in case too long shifts were executed. Thus, the latter scheme also produces more accurate results than the canonic add/shift implementation.

6. JPEG ENCODER IMPLEMENTATION SCHEMES AND RESULTS

Different implementation schemes, each satisfying the constraints defined in Section 3, were evaluated. These multiplierless implementations were used to compress a set of 23 images which feature different degrees of image complexity. The test images are full color, 24 bits per pixel, with a spatial resolution of 640x480 pixels (VGA format). After encoding with each of the different schemes (including SIMI), the images were all decompressed with a floating point JPEG decoder, for peak signal to noise (PSNR) calculation and for visual evaluation of the reconstructed image quality.

Each of the schemes presents its own advantages; the following paragraphs report the results of two implementation structures. The first one requires a complex control unit, while the second approach uses the simplest. The signal data wordlength was 16 bits in both cases.

1) For the first implementation scheme, each coefficient was selected as to require the least number of add/shift iterations (i.e., each binary number should have the least number of 1's). The results reported in Table 1, 2, and 3, correspond to this approximation scheme. Since the maximum number of add/shift is not the same for all the coefficients, this implementation requires a control unit of higher complexity. For this scheme the average peak signal to noise ratio (PSNR) over the 23 test images drops by 0.225 dB, for a practically identical compression ratio per image. Figure 3c) shows a region of an image with such a global drop in the PSNR value.



a) b) c) d)

Figure 3. Reconstructed images, compression ratio= 22 in all cases. a) Original; b) SIMI: 35.478 dB; c) Scheme 1: 35.249 dB; d) Scheme 2: 35.467 dB

2) From Table 3 one can observe that the maximum number of add/shift operations, as well as the maximum individual shift of the multiplierless system are required during the quantization. In both cases, this number is equal to 6. Thus, the control unit could be largely simplified if all the pseudo-multiplications were implemented using a fixed number of 6 add/shift operations (along with the necessary data or instruction padding for the cases where the coefficient is more accurately approximated by using less than 6 non-zero bits, e.g., when $\hat{a}$=0.5, or $\hat{a}$=0.25). Furthermore, the fact of loosening the constraints on these two parameters, allows the approximation of the a coefficients with a higher accuracy.

The benefit of improving the precision of the approximations manifests itself on the resulting PSNR values. For this implementation scheme, the average drop of

Original coefficients	Multiplierless approximation	Binary representation	Number of add/shift	Maximum single shift	Relative error
0.299	0.29687500	0.0100110000	3	3	0.71%
0.587	0.58593750	0.1001011000	4	3	0.18%
0.114	0.11328125	0.0001110100	4	4	0.63%
0.169	0.16796875	0.0010101100	4	3	0.61%
0.331	0.32812500	0.0101010000	3	2	0.87%
0.500	0.50000000	0.1000000000	1	1	0.00%
0.419	0.42187500	0.0110110000	4	2	0.69%
0.081	0.08105469	0.0001010011	4	4	0.07%

Table 1. Approximation of the coefficients for the RGB to YCbCr conversion

Original coefficients	Multiplierless approximation	Binary representation	Number of add/shift	Maximum single shift	Relative error
0.7071	0.70312500	0.10110100	4	2	0.56%
0.2706	0.26953125	0.01000101	3	4	0.39%
0.6533	0.65625000	0.10101000	3	2	0.45%
0.3827	0.38281250	0.01100010	3	4	0.03%

Table 2. Approximation of the coefficients for the DCT computation

Number of add/shift	Maximum single shift					
	1	2	3	4	5	6
1	-	1	2	3	-	-
2	1	2	4	6	4	4
3	-	3	8	28	9	-
4	-	3	12	10	4	1
5	-	5	1	8	4	-
6	-	1	-	4	-	-

Table 3. Frequency of occurrence of the different configurations for the approximated normalization coefficients

the PSNR over the test image set is of a negligible 0.018 dB, for a virtually identical compression ratio per image; as before, with respect to the results produced by a SIMI. Figure 3d) shows the image quality of this approximation scheme.

6.1.1. Compressing with multiple image quality

Due to the *fixed* approximation of the quantization values reported in Table 3, this multiplierless implementation produces only one degree of image compression, i.e., always with an excellent reconstruction quality (due to the ½ scaling made on the quantization coefficients) which is just what might be needed in many systems. Nevertheless, other applications could require the possibility of allowing the user or the system to execute compression with several degrees of image quality. In this case, the system has to include different sets of approximated values $\hat{a}$; each set, corresponding to a different scaling (and thus of image reconstruction quality) of the normalization matrix.

7. CONCLUSIONS

In this paper, a multiplierless implementation scheme for an industry standard image compression algorithm has been reported. The arithmetic operations required in this implementation have been reduced to simple additions and very short shifts. Two approximation schemes were reported, the more accurate of them, produces practically the same results as those obtained by a system using standard multiplications.

These results demonstrate that a relatively complex image processing algorithm can be implemented within a system of limited hardware resources, and/or in portable applications where silicon area and power consumption are sensitive design parameters.

8. ACKNOWLEDGEMENTS

This work was supported by the Swiss Federal Office for Education and Science under Grant OFES C97.0050 (COST 254 Research Project).

9. REFERENCES

[1] B. W. Wah and Z. Wu, "Discrete lagrangian methods for designing multiplierless two-channel PR-LP filter banks", *J. of VLSI Signal Processing*, Kluwer Academic Press, Vol. 21, No. 2, June 1999, pp. 131–150.

[2] S. Samadi, A. Nishihara, and N. Fujii, "Modular array structures for design and multiplierless realization of two-dimensional linear phase FIR digital filters", *IEICE Trans. Fundamental of Elect., Comm., and Comp. Sc.*, Vol. E80-A, No. 4, April 1997, pp. 722–736.

[3] A. N. Akansu, "Multiplierless PR quadrature mirror filters for subband image coding", *IEEE Trans. on Image Processing*, Vol. 5, No. 9, Sept. 1996, pp. 1359–1363.

[4] ITU-T Recommendation T.81, Digital Compression and coding of continuous-tone still images, September, 1992.

[5] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images", *Trans. of the IEICE*, Vol. E71, No. 11, Nov. 1988, pp. 1095–1097.