JPEG quantisation requires bit-shifts only

P.A.M. Oliveira, R.S. Oliveira, R.J. Cintra✉, F.M. Bayer and A. Madanayake

A scheme for low power consumption JPEG quantisation is introduced. Multiplications and additions are not required – only bit-shifting operations. A fully multiplierless JPEG-like image encoding framework is also proposed, using a discrete cosine transform approximation. Simulations demonstrate that the introduced quantisation scheme produces a very low image degradation. The proposed architecture is also physically realised in FPGA technology.

Introduction: Consumer electronics fosters digital convergence and high performance devices at limited power resources as exemplified in high-definition TV standards and mobile devices [1]. Similarly, wireless sensor networks demand low-cost implementations capable of reduced energy consumption [2]. Such applications require increasing miniaturisation, low power consumption, and reduced amounts of data storage/transmission. Indeed, image and video compression is still a big challenge for fast algorithm designers. In terms of image compression, the JPEG standard is commonly adopted in most schemes [3]. Its computational cost is in great part due to transform and quantisation steps, which often require a large amount of multiplications. Multiplications have a high computational cost and may be a hindrance to the dissemination of low-power and real-time systems.

In this Letter, we propose a significantly low-complexity, JPEG-compatible quantisation scheme requiring neither multiplications nor additions – only bit-shifting operations are needed. The proposed low-complexity quantisation block can be combined with discrete cosine transform (DCT) approximations [4] to reduce the complexity of the transform-quantisation pair of the JPEG encoding/decoding algorithm. Computational experiments indicate that the resulting encoded images exhibit nearly identical image quality. Combinations of the DCT, a DCT approximation, the usual, and the proposed quantisation schemes are implemented using FPGA technology. Obtained measurements also highlight the good performance of the proposed quantisation scheme.

JPEG encoding and decoding: After the image preprocessing stage, which may comprise space colour transformation, chroma subsampling, and unbiasing [3], an $M \times N$ input image is subdivided into 8×8 size blocks $\boldsymbol{I}_{i,j}$, $i = 1, 2, \ldots, M/8$, $j = 1, 2, \ldots, N/8$. Each block is submitted to the 2D DCT, $\boldsymbol{H}_{i,j} = \boldsymbol{C} \cdot \boldsymbol{I}_{i,j} \cdot \boldsymbol{C}^{\top}$, where $\boldsymbol{C}$ is the DCT matrix and $\boldsymbol{H}_{i,j}$ is the associate DCT-domain image block. Then the DCT-domain coefficients are quantised according to Wallace [3]:

$$\boldsymbol{H}_{i,j}^{(\text{quant})} = \text{round}\big(\boldsymbol{H}_{i,j} \oslash \boldsymbol{Q}\big) \tag{1}$$

where round$(\cdot)$ is the rounding function, $\oslash$ denotes the element-wise division, $\boldsymbol{Q} = \lfloor (S \cdot \boldsymbol{Q}_0 + 50)/100 \rfloor$ is the quantisation matrix, $\lfloor \cdot \rfloor$ denotes the floor function, $\boldsymbol{Q}_0$ and S are given by

$$\boldsymbol{Q}_0 = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 84 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix},$$

$$S = \begin{cases} 5000/QF, & \text{if } QF < 50 \\ 200 - 2QF, & \text{otherwise} \end{cases}$$

and QF is the quality factor [3]. All matrix operators are taken element-wise. If $QF = 50$, then $\boldsymbol{Q} = \boldsymbol{Q}_0$. Decreasing values of QF lead to higher compression ratios (complete image destruction at $QF = 0$); whereas increasing values produce lower compression ratios (best possible quality at $QF = 100$). After quantisation, the DC coefficient of each block is submitted to differential encoding; whereas the remaining 63 coefficients are ordered according to zig-zag scan pattern for Huffman encoding [3]. The decoding operation comprises the dequantisation step given by:

$$\widehat{\boldsymbol{H}}_{i,j} = \boldsymbol{H}_{i,j}^{(\text{quant})} \odot \boldsymbol{Q} \tag{2}$$

where $\widehat{\boldsymbol{H}}_{i,j}$ is the dequantised DCT-domain coefficients and $\odot$ denotes the element-wise multiplication. The coefficients $\widehat{\boldsymbol{H}}_{i,j}$ are not expected to be precisely recovered due to the round operation in the quantisation step. Subsequently, the inverse 2D DCT transform is performed according to $\widehat{\boldsymbol{I}}_{i,j} = \boldsymbol{C}^{\top} \cdot \widehat{\boldsymbol{H}}_{i,j} \cdot \boldsymbol{C}$. The full image is reconstructed after block concatenation of the recovered blocks and suitable post-processing. The whole process involves a large number of multiplications, which are concentrated in the DCT and quantisation phases. We aim at significantly reducing the number of multiplications while preserving high compressed image quality.

Approximate quantisation: The quantisation/dequantisation stages [cf. (1) and (2)] demand for each output image pixel two multiplications and one call of the rounding operation. We advance an approximation for the quantisation matrix based on power-of-two integers. Such an approximation is sought to be free of both multiplication and addition operations – only bit-shifting operations are required. Thus, we introduce an approximation for $\boldsymbol{Q}$ given by:

$$\widetilde{\boldsymbol{Q}} = \text{np2}(\boldsymbol{Q})$$

where $\text{np2}(x) = 2^{\text{round}(\log_2 x)}$, $x \in \mathbb{R}$, is the nearest power of two function operated element-wise. For the JPEG quantisation table $\boldsymbol{Q}_0$, we obtain the following approximate matrix:

$$\widetilde{\boldsymbol{Q}}_0 = \begin{bmatrix} 16 & 8 & 8 & 16 & 32 & 32 & 64 & 64 \\ 16 & 16 & 16 & 16 & 32 & 64 & 64 & 64 \\ 16 & 16 & 16 & 24 & 32 & 64 & 64 & 64 \\ 16 & 16 & 16 & 32 & 64 & 64 & 64 & 64 \\ 16 & 16 & 32 & 64 & 64 & 128 & 128 & 64 \\ 32 & 32 & 64 & 64 & 64 & 128 & 128 & 64 \\ 64 & 64 & 64 & 64 & 128 & 128 & 128 & 128 \\ 64 & 64 & 64 & 128 & 128 & 128 & 128 & 128 \end{bmatrix}$$

The above matrix can be immediately embedded into the JPEG codec without any further modifications.

Multiplierless JPEG transform-quantisation pair: Although the approximate quantisation only employs bit-shifting operations, the DCT transform stage still requires multiplications, additions, and floating-point arithmetic. However, such computational load can be drastically reduced by considering multiplierless integer DCT approximations [4, 5]. In this Letter, we do not aim at designing a new approximate DCT; but considering DCT approximations as a venue to further reduce the complexity of the JPEG codec in combination with the proposed quantisation scheme. Thus we separate the rounded DCT (RDCT) [4], which presents a good trade-off between computational complexity and coding performance. The RDCT is given by the product of a diagonal matrix ($\boldsymbol{S}$) of irrational coefficients and an integer matrix $\widehat{\boldsymbol{C}} = \boldsymbol{S} \cdot \boldsymbol{C}_0$, where $\boldsymbol{S} = \text{diag}(1/\sqrt{8}, 1/\sqrt{6}, 1/2, 1/\sqrt{6}, 1/\sqrt{8}, 1/\sqrt{6}, 1/2, 1/\sqrt{6})$ and $\boldsymbol{C}_0$ is the low-complexity, integer matrix with entries in $\{-1, 0, +1\}$ [4]. The multiplicative nature of $\boldsymbol{S}$ can be eliminated by merging its entries into forward and inverse quantisation matrices, $\widehat{\boldsymbol{Q}}_{\text{f}}$ and $\widehat{\boldsymbol{Q}}_{\text{i}}$, respectively:

$$\widehat{\boldsymbol{Q}}_{\text{f}} = \text{np2}\big((\boldsymbol{s} \cdot \boldsymbol{s}^{\top}) \oslash \boldsymbol{Q}\big) \text{ and } \widehat{\boldsymbol{Q}}_{i} = \text{np2}\big((\boldsymbol{s} \cdot \boldsymbol{s}^{\top}) \odot \boldsymbol{Q}\big)$$

where $\boldsymbol{s}$ is the column-vector with the elements of the main diagonal of $\boldsymbol{S}$. Hence, the low-complexity combined transform and quantisation pair is furnished by:

$$\boldsymbol{H}_{i,j}^{(\text{quant})} = \big(\boldsymbol{C}_0 \cdot \boldsymbol{I}_{i,j} \cdot \boldsymbol{C}_0^{\top}\big) \oslash \widehat{\boldsymbol{Q}}_{f}, \quad \text{(encoding)} \tag{3}$$

$$\widehat{\boldsymbol{I}}_{i,j} = \boldsymbol{C}_0^{\top} \cdot \Big(\boldsymbol{H}_{i,j}^{(\text{quant})} \odot \widehat{\boldsymbol{Q}}_{i}\Big) \cdot \boldsymbol{C}_0, \quad \text{(decoding)} \tag{4}$$

Thus, the JPEG codec can be modified by considering approximations under three scenarios: (i) approximating the DCT matrix only; (ii) approximating the quantisation matrix only; and (iii) approximating both the DCT and the quantisation matrices.

Table 1 shows the arithmetic cost for encoding and decoding a single 8×8 image block depending on the computing stage to be approximated. For the fully approximate scenario, i.e. both transform and quantisation stages are submitted to approximations, we obtain a

multiplierless transform-and-quantisation pair with arithmetic operation savings of 24.1% in additions when compared to the standard JPEG. The extra bit-shifting operations can be seamlessly implemented in hardware with virtually no computational cost. Moreover, the proposed quantisation also completely eliminates the need for the rounding function.

Table 1: Arithmetic cost comparison

Approximate block	Mult.	Add.	Bit-shift	round(·)
None (standard JPEG)	288	928	0	64
Transform (RDCT [4])	128	704	0	64
Quantisation (proposed)	160	928	128	0
Transf. and Quant. (proposed)	0	704	128	0

Image compression: To assess the proposed quantisation scheme, we embedded it into the JPEG codec and processed a set of 45 512 × 512 8-bit greyscale images obtained from a standard public image bank [6]. We adopted the three approximation scenarios discussed in the previous section as well as the unmodified JPEG standard. The peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) [7] were employed for image quality assessment between original and reconstructed images. For each value of QF, we considered the average quality measurements values and Fig. 1 shows the results. The proposed quantisation performed almost identically in comparison with its exact counterparts; for exact or approximate transforms. For qualitative purposes, Fig. 2 shows compressed images under the discussed scenarios for $QF = 50$; resulting images are almost indistinguishable. In summary, the proposed quantisation has a vastly lower computational cost and still maintains high image quality.

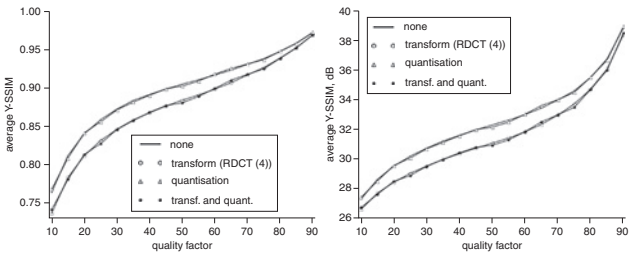

Fig. 1 *SSIM and PSNR metrics in terms of QF*

Fig. 2 *Compressed Lena image for QF = 50 using*
a Standard JPEG
b Approximate DCT [5]
c Proposed quantisation
d Approximate transform and proposed quantisation

FPGA implementation: The encoding process was implemented on an FPGA chip for each scenario using the Xilinx Virtex-6 XC6VLX240T 1FFG1156 device. We adopted $QF = 50$. Considering hardware co-simulation, the FPGA realisation was tested with 80,000 random 8-point input test vectors. The test vectors were generated from within the MATLAB environment and, using JTAG-based hardware co-simulation, routed to the physical FPGA device where each algorithm was realised in the reconfigurable logic fabric. Then the computational results obtained from the FPGA algorithm implementations were routed back to MATLAB memory space.

The metrics employed to evaluated the FPGA implementations were: configurable logic blocks (CLB), flip-flop (FF) count, and critical path delay (T_{cpd}). The maximum operating frequency was determined by the critical path delay as $F_{\max} = (T_{\text{cpd}})^{-1}$. Using the Xilinx XPower analyser, we estimated the static (Q_p) and dynamic power (D_p) consumption. In addition, we calculated area-time (AT) and area-time-square (AT^2) figures of merit, where area (A) was measured as the CLB count and time (T), as the critical path delay. The results are shown in Table 2. Approximating only the quantisation block effected an improvement of 80.80 and 89.34% in terms of AT and AT^2, respectively, from the case where none of the blocks were approximated. However, approximating both the transform and quantisation blocks resulted in an even greater improvement of 96.74 and 98.60% in terms of AT and AT^2, respectively, from the standard JPEG scheme. Moreover, the fully approximated scheme allows the system to work at a 132% higher frequency than the standard JPEG scheme.

Table 2: Hardware resource consumption and power consumption

Measure	None (JPEG)	Transform (RDCT [4])	Quantisation (proposed)	Transf. and Quant. (proposed)
CLB	8096	3084	2797	615
FF	35,172	7747	12,539	2248
T_{cpd} (ns)	6.30	6.20	3.50	2.70
$F_{\max}$(MHz)	159	161	286	370
D_p(mW/GHz)	5.758	1.928	10.570	4.207
Q_p (W)	3.450	3.431	3.518	3.471
AT	51,005	19,120	9790	1661
AT^2	321,330	118,549	34,263	4483

Conclusion: In this Letter, we introduced a JPEG approximate quantisation step that demands only bit-shifting operations. We also devised a multiplication-free transform-and-quantisation stage for the JPEG algorithm by utilising an approximate DCT. Computational simulations show that the image degradation introduced by the approximate quantisation is negligible. In the FPGA experiments, it was shown that the most efficient scheme was the one approximating both transform and quantisation blocks. This framework could be successfully implemented in systems which demand real-time processing and very low energy consumption.

Acknowledgment: We thank CNPq, FAPERGS, and The College of Engineering at UA for the partial financial support.

Submitted: *27 November 2016* E-first: *31 March 2017*
doi: 10.1049/el.2016.4342

P.A.M. Oliveira (*Multimedia Communications and Signal Processing, FAU Erlangen-Nuremberg, Erlangen, BY, Germany*)

R.S. Oliveira and R.J. Cintra (*Signal Processing Group, Universidade Federal de Pernambuco, PE, Brazil*)

✉ E-mail: rjdsc@stat.ufpe.org

F.M. Bayer (*Departamento de Estatstica and LACESM, Universidade Federal de Santa Maria, RS, Brazil*)

A. Madanayake (*ECE, The University of Akron, Akron, OH, USA*)

References

1 Wahid, K., Ko, S.-B., and Teng, D.: 'Efficient hardware implementation of an image compressor for wireless capsule endoscopy applications'. IEEE Int. Joint Conf. on Neural Networks (IJCNN), Hong Kong, HKG, June 2008, pp. 2761–2765
2 Kouadria, N., Doghmane, N., Messadeg, D., and Harize, S.: 'Low complexity DCT for image compression in wireless visual sensor networks', *Electron. Lett.*, 2013, **49**, (24), pp. 1531–1532
3 Wallace, G.K.: 'The JPEG still picture compression standard', *IEEE Trans. Consum. Electron.*, 2004, **38**, (1), pp. xviii–xxxiv
4 Cintra, R.J., and Bayer, F.M.: 'A DCT approximation for image compression', *Signal Process. Lett.*, 2011, **10**, pp. 579–582
5 Cintra, R.J., Bayer, F.M., and Tablada, C.J.: 'Low-complexity 8-point DCT approximations based on integer functions', *Signal Process..*, 2014, **99**, pp. 201–214
6 The USC-SIPI Image Database, University of Southern California, Signal and Image Processing Institute. Available at http://sipi.use.edu/database, accessed January 2016
7 Wang, Z., Bovik, A.C., Sheikh, H.R., and Simoncelli, E.P.: 'Image quality assessment: from error visibility to structural similarity', *IEEE Trans. Image Process.*, 2004, **13**, (4), pp. 600–612