J Sign Process Syst (2012) 66:135–139
DOI 10.1007/s11265-011-0590-1

An Improved Floating-to-Fixed-Point Conversion Scheme for DCT Quantization Algorithm

Lin Wang · Fuliang Yin · Zhe Chen

Received: 29 March 2010 / Revised: 11 December 2010 / Accepted: 25 April 2011 / Published online: 17 May 2011

Abstract Conventional fixed-point implementation of the DCT coefficients quantization algorithm in video compression may result in deteriorated image quality. The paper investigates this problem and proposes an improved floating-to-fixed-point conversion scheme. With a proper scaling factor and a new-established look-up table, the proposed fixed-point scheme can obtain bit-wise consistence to the floating-point realization. Experimental results verify the validity of the proposed method.

Keywords Floating-to-fixed-point conversion · Discrete cosine transform · Quantization · Video compression

1 Introduction

Digital signal processing applications are specified with floating-point type but they are usually implemented in embedded systems with fixed-point arithmetic to minimize cost and power consumption [1]. Thus floating-point to fixed-point conversion is required in order to achieve acceptable levels of performance and cost. This is particularly crucial in real-time applications that have strict timeline requirement and processing throughput and latency. Some algorithms in video compression are typically designed with floating-point operations, such as discrete cosine transform (DCT) and coefficients quantization etc. It's necessary to convert these floating-point algorithms to fixed-point ones before realizing them in fixed-point hardware efficiently.

Floating-to-fixed-point conversion is relatively mature since it has appeared for decades [2, 3]. Some general methods have been formed and widely used. However, in some specific application, conventional methods can not provide sufficient precision resulting in large errors. The paper investigates such a problem in the floating-to-fixed-point conversion of the DCT quantization algorithm in video compression, and proposes an improved solution.

The organization of the paper is as follows. Section 2 introduces the conventional fixed-point realization of the DCT quantization algorithm. Section 3 analyzes the floating-to-fixed-point conversion error and proposes a possible solution. Experiments results are given in Section 4. Finally conclusions are drawn in Section 5.

2 Fixed-Point Realization of DCT Coefficients Quantization

DCT, quantization, inverse quantization, and inverse DCT are essential processing steps in video coding standards [4, 5]. DCT has become a standard method in image and video compression. Typically an image is divided into 8×8-pixels blocks, which are each

This work is supported by the Specialized Research Fund for the Doctoral Program of Higher Education of China (200801410015).

L. Wang (✉) · F. Yin · Z. Chen
School of Electronic and Information Engineering, Dalian University of Technology, Dalian, 116023, People's Republic of China
e-mail: wanglin_2k@sina.com

F. Yin
e-mail: flyin@dlut.edu.cn

Z. Chen
e-mail: eeyin@dlut.edu.cn

transformed with 64 transform coefficients. The 8×8 DCT coefficients of each block have to be quantized before entropy coding. Quantization is a many-to-one mapping where the 64 coefficients of an 8×8 block is divided by the quantization step; then the result is rounded. Since a large number of coefficients become zero after quantization, compression is achieved. Quantization is a crucial step in that it allows to reduce the accuracy with which the DCT coefficients are presented, opening the door to different rate quality trade-offs. In principle, the step size could be computed and optimized for each 8×8 block so as to achieve minimum distortion for a given target bit rate.

In various video compression standards, different quantization algorithms are employed. For simplicity and without loss of generality, for an original value F and a quantization step P, the quantized value C is expressed as:

$$C = \text{sign}(F) \cdot \text{round}\left(\frac{|F|}{P}\right) \tag{1}$$

where $|\cdot|$ denotes the absolute value of the argument, sign$(\cdot)$ denotes the sign of it, and round$(\cdot)$ denotes rounding to the nearest integer. And the reconstructed (dequantized) value $\hat{F}$ after inverse quantization is

$$\tilde{F} = \text{sign}(F) \cdot C \cdot P \tag{2}$$

Division arithmetic used in quantization involves floating-point operation. Generally, floating-point division can be implemented in fixed-pointed form as below.

From Eq. 1, we get

$$C = \text{sign}(F) \cdot \left\lfloor \frac{|F| + P/2}{P} \right\rfloor = \text{sign}(F) \cdot \left\lfloor \frac{B}{P} \right\rfloor \tag{3}$$

where $\lfloor \cdot \rfloor$ denotes rounding towards zero, and

$$B = |F| + P/2 \tag{4}$$

The fixed-point version of C is

$$\begin{aligned} \hat{C} &= \text{sign}(F) \cdot \left\lfloor \frac{B \cdot (2^Q/P)}{2^Q} \right\rfloor \\ &= \text{sign}(F) \cdot \{(B \cdot T_P) >> Q\} \end{aligned} \tag{5}$$

where Q is the scale factor, $(\cdot) >> Q$ denotes right shifting Q bits of the argument, and

$$T_P = \lfloor 2^Q/P \rfloor \tag{6}$$

can be calculated in advance and stored in a table. So, with a scale factor Q and a look-up table T_P, the division operation can be replaced by multiplication and shift operations.

Table 1 Fixed-point realization of the quantization algorithm

Example	F	P	C	Q	$\hat{C}$	B
1	144	12	12	10	12	150
2	138	4	35	8	35	140
3	138	12	12	1∼32	11	144

This floating-to-fixed-point conversion scheme works well in most cases. However, in some situation, it fails to provide enough precision as the floating-point one does. Table 1 gives some examples of the fixed-point realization of the quantization algorithm. In the first two examples, $\hat{C} = C$ can be guaranteed by choosing a proper scale factor Q. In the third example where Q is a random number in the range [1, 32], a floating-to-fixed-point conversion error with $\hat{C} = C - 1$. What's more, it is noticed that $\hat{C} = C - 1$ always hold no matter how large Q is. Although it is a small deviation, the error will be enlarged after inverse quantization. In the third example, the dequantized values by Eq. 2 are 144 and 132, respectively for the floating-point algorithm and the fixed-point one: an evident gap between them. It will deteriorate the reconstructed image quality greatly if the floating-to-fixed-point error happens frequently. Some examples in Section 4 will demonstrate this.

3 Improved Floating-to-Fixed-Point Conversion Method

In this section, the reason for the floating-to-fixed-point conversion error is investigated and an improved method is proposed based on it. Without loss of generality, we suppose the original value F is positive.

Comparing Eqs. 3 and 5, it is found that the difference between C and $\hat{C}$ depends on the values of (B/P) and $\lfloor B/P \rfloor$. Since $(B/P) \geq \lfloor B/P \rfloor$, we discuss from two aspects respectively: $(B/P) > \lfloor B/P \rfloor$ and $(B/P) = \lfloor B/P \rfloor$.

3.1 $(B/P) > \lfloor B/P \rfloor$

This case corresponds to the first example in Section 2.

It can be obtained that

$$\frac{B}{P} > \left\lfloor \frac{B}{P} \right\rfloor \geq \left\lfloor \frac{B \cdot \lfloor 2^Q/P \rfloor}{2^Q} \right\rfloor \tag{7}$$

Furthermore, the error between (B/P) and $\lfloor B/P \rfloor$ lies in the range

$$\frac{1}{2P} \leq \frac{B}{P} - \left\lfloor \frac{B}{P} \right\rfloor < 1 \tag{8}$$

and the error between (B/P) and $\left\lfloor \frac{B\cdot\lfloor 2^Q/P\rfloor}{2^Q}\right\rfloor$ is

$$\frac{B}{2^Q P} \le \frac{B}{P} - \left\lfloor \frac{B\cdot\lfloor 2^Q/P\rfloor}{2^Q}\right\rfloor < \frac{B}{2^Q} \tag{9}$$

From Eqs. 8 and 9, it is observed that

$$\left\lfloor \frac{B}{P}\right\rfloor \le \left\lfloor \frac{B\cdot\lfloor 2^Q/P\rfloor}{2^Q}\right\rfloor \tag{10}$$

when

$$\frac{1}{2P} \ge \frac{B}{2^Q} \tag{11}$$

From Eqs. 7 and 10, equation $\lfloor \frac{B}{P}\rfloor = \left\lfloor \frac{B\cdot\lfloor 2^Q/P\rfloor}{2^Q}\right\rfloor$, i.e. $C=\hat{C}$, will hold once Eq. 11 is satisfied, i.e.

$$2^Q \ge 2BP \tag{12}$$

In this way, by choosing an appropriate value of the scale factor Q, $C=\hat{C}$ can be guaranteed.

3.2 $(B/P)=\lfloor B/P\rfloor$

We discuss from two aspects: $(2^Q/P)=\lfloor 2^Q/P\rfloor$ and $(2^Q/P)>\lfloor 2^Q/P\rfloor$.

(1) $(2^Q/P)=\lfloor 2^Q/P\rfloor$
Since

$$\hat{C}=\left\lfloor \frac{B\cdot\lfloor 2^Q/P\rfloor}{2^Q}\right\rfloor = \left\lfloor \frac{B\cdot(2^Q/P)}{2^Q}\right\rfloor = C$$

$C=\hat{C}$ always holds. This corresponds to the second example in Section 2.

(2) $(2^Q/P)>\lfloor 2^Q/P\rfloor$
Since $(B/P)=\lfloor B/P\rfloor$, $C=\lfloor B/P\rfloor=(B/P)$ is always an integer.
Since $(2^Q/P)>\lfloor 2^Q/P\rfloor$, $\hat{C}<C$ always holds.
On the other hand, $\hat{C}$ is also an integer, thus $\hat{C}=C-1$ always holds no matter how large Q is. This corresponds to the third example in Section 2.

To cope with the problem above, a new floating-to-fixed-point conversion scheme is proposed which employs a new established table $\hat{T}_P$ in lieu of the original T_P in Eq. 6.

$$\hat{T}_P = \begin{cases} \lfloor 2^Q/P\rfloor + 1, & \text{if } \lfloor 2^Q/P\rfloor < (2^Q/P) \\ \lfloor 2^Q/P\rfloor, & \text{if } \lfloor 2^Q/P\rfloor = (2^Q/P) \end{cases} \tag{13}$$

Consequently, the new fixed-point version of C is

$$\tilde{C} = \begin{cases} \left\lfloor \frac{B\cdot(\lfloor 2^Q/P\rfloor+1)}{2^Q}\right\rfloor, & \text{if } \lfloor 2^Q/P\rfloor < (2^Q/P) \\ \left\lfloor \frac{B\cdot(2^Q/P)}{2^Q}\right\rfloor, & \text{if } \lfloor 2^Q/P\rfloor = (2^Q/P) \end{cases} \tag{14}$$

Comparing Eqs. 3 and 14, it is found that $C=\tilde{C}$, when $\lfloor 2^Q/P\rfloor=(2^Q/P)$; and $C<\tilde{C}$, when $\lfloor 2^Q/P\rfloor < (2^Q/P)$. Now we discuss the second case $\lfloor 2^Q/P\rfloor < (2^Q/P)$.

Similar to the analysis in Section 3.1, we get

$$\left\lfloor \frac{B}{P}\right\rfloor = \frac{B}{P} \le \left\lfloor \frac{B\cdot\left(\lfloor 2^Q/P\rfloor+1\right)}{2^Q}\right\rfloor \le \left\lfloor \frac{B}{P}\right\rfloor + 1 = \frac{B}{P}+1 \tag{15}$$

Thus $\lfloor \frac{B}{P}\rfloor = \left\lfloor \frac{B\cdot(\lfloor 2^Q/P\rfloor+1)}{2^Q}\right\rfloor$ will hold as long as the condition

$$\left\lfloor \frac{B\cdot\left(\lfloor 2^Q/P\rfloor+1\right)}{2^Q}\right\rfloor < \frac{B}{P}+1 \tag{16}$$

is satisfied.

Furthermore, the error between $\left\lfloor \frac{B\cdot(\lfloor 2^Q/P\rfloor+1)}{2^Q}\right\rfloor$ and (B/P) is

$$\frac{B}{2^Q P} < \left\lfloor \frac{B\cdot\left(\lfloor 2^Q/P\rfloor+1\right)}{2^Q}\right\rfloor - \left(\frac{B}{P}\right) < \frac{B}{2^Q} \tag{17}$$

Thus, from Eqs. 16 and 17, $\lfloor \frac{B}{P}\rfloor = \left\lfloor \frac{B\cdot(\lfloor 2^Q/P\rfloor+1)}{2^Q}\right\rfloor$ will hold when

$$2^Q > B \tag{18}$$

By choosing an appropriate value of the scale factor Q, we can ensure that $C=\tilde{C}$.

In summary, based on the discussion in Sections 3.1 and 3.2, the floating-to-fixed-point conversion error of the quantization algorithm will be eliminated with a newly established table by Eq. 13, and two conditions in Eqs. 12 and 18 satisfied. Moreover, since $BP\ge B$, satisfying Eq. 12, $2^Q>2BP$, is sufficient for both conditions. It defines the relationship among the original value B, quantization step P, and scale factor Q.

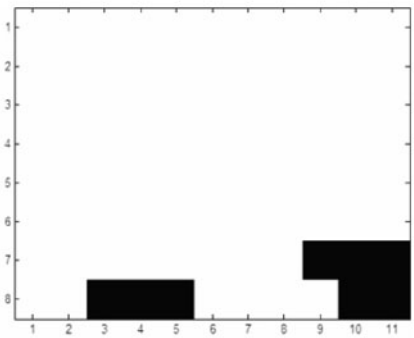
In video compression, the dynamic ranges of the DCT coefficients and the quantization step is not very large, thus it is feasible to choose an appropriate scale factor Q. For example, in MPEG-4 video compression standard, the dynamic ranges of the DCT coefficients and the quantization step are $B<2{,}048$, $P<48$, hence setting $Q=18$ can satisfies Eq. 12. In this way, the new fixed-point version of C in the third example is $\tilde{C}=12=C$.

4 Experimental Results

In this experiment, the conventional (cf. Eq. 5) and the proposed floating-to-fixed-point conversion schemes

(a) Floating-point algorithm

(b) Conventional fixed-point algorithm



(c) Quantization error distribution by the conventional fixed-point algorithm

(d) Improved fixed-point algorithm

Figure 1 Comparison of the reconstructed images by the floating-point algorithm, the conventional fixed-point algorithm, and the improved fixed-point algorithm.

are compared when developing a fixed-point DCT coefficients quantization algorithm in MPEG-4 video compression standard. A test sequence "Mthr_dotr" of size QCIF (resolution 176×144) is used in the experiment. Both DCT coefficients quantization and DC/AC prediction are realized in the floating-point form, the conventional fixed-point form, and the improved fixed-point form respectively. We set $Q = 18$ in the later two fixed-point algorithms. One typical reconstructed I-frame image after encoding with the three algorithms, respectively, is shown in Fig. 1.

Comparing Fig. 1a and b, degraded image quality can be observed for the conventional fixed-point algorithm. This is due to the floating-to-fixed-point conversion error. Figure 1c depicts the quantization error distribution of the DC coefficients of the U (chrominance) blocks in the image, with the horizontal axis denoting horizontal macroblocks and the vertical axis denoting vertical ones. In Fig. 1c, white color denotes no quantization error occurring and dark color marks quantization error. Similar quantization errors also occur for Y and U data blocks. Furthermore, the quantization errors may transfer to subsequent macroblocks and frames by DC/AC prediction and inter-frame prediction. Finally, the accumulated errors result in the deteriorated image: the image color at the bottom right is obviously changed. In particular, the error of the DC coefficients in U and V blocks lead to the color change. Although this phenomenon does not happen very often, it degrades the image quality greatly, which is unacceptable. The reconstructed image with the improved fixed-point algorithm is shown in Fig. 1d, where no color change is observed. Furthermore, bit-wise identity is obtained between the floating-point algorithm and the improved fixed-point one.

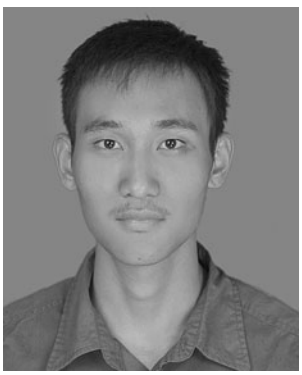
5 Conclusions

The paper investigates the fixed-point realization of the DCT quantization algorithm in video compression, where conventional floating-to-fixed-point conversion scheme can not provide enough precision and results in degraded reconstructed image quality. The reasons for the floating-to-fixed-point conversion errors are analyzed and a new floating-to-fixed-point conversion scheme is proposed based on it. With a proper scaling factor and a new-established look-up table, the proposed method can obtain bit-wise consistence to the floating-point one.

References

1. Oppenheim, A. V., & Schafer, R. W. (1975). *Digital signal processing*. New Jersey: Prentice-Hall.
2. Kim, S., Kum, K. I., & Sung, W. (1998). Fixed-point optimization utility for C and C++ based digital signal processing programs. *IEEE Transactions on Circuits and Systems Part II, 45*(11), 1455–1464.
3. Menard, D., Chillet, D., & Sentieys, O. (2006). Floating-to-fixed-point conversion for digital signal processor. *EURASIP Journal on Applied Signal Processing, 2006*(1), 1–19.
4. Rijkse, K. (1996). H. 263: Video coding for low-bit-rate communication. *IEEE Communications Magazine, 34*(12), 42–45.
5. Richardson, I. E. G. (2003). *H. 264 and MPEG-4 video compression: Video coding for next-generation multimedia*. New York: Wiley.

Fuliang Yin was born in Fushun city, Liaoning province, China, in 1962. He received the B.S. degree in electronic engineering and the M.S. degree in communications and electronic systems from Dalian University of Technology (DUT), Dalian, China, in 1984 and 1987, respectively. He joined the Department of Electronic Engineering, DUT, as a Lecture in 1987 and became an Associate Professor in 1991. He has been a Professor at DUT since 1994, and the Dean of the School of Electronic and Information Engineering of DUT since 2000. His research interests include digital signal processing, speech processing, image processing, broadband wireless communication, and integrated circuit design.



Lin Wang was born in Anhui, China, in 1981. He received the B.S. degree in electronic engineering from Tianjin University, Tianjin, China, in 2003; and the Ph.D degree in signal processing from Dalian University of Technology, Dalian, China, in 2010. He has been a visiting fellow with the Institute for Microstructural Sciences, National Research Council Canada, for thirteen months, from 2008 to 2009. His research interests include video and audio compression, blind source separation, 3D audio processing.

Zhe Chen was born in Heilongjiang province, China, in 1975. He received the B.S. degree, the M.S. degree in signal and information processing, and the Ph.D. degree in signal and information processing from Dalian University of Technology (DUT), Dalian, China, in 1996, 1999 and 2003 respectively. He joined the Department of Electronic Engineering, DUT, as a Lecture in 2002, and became an Associate Professor in 2006. His research interests include stochastic signal processing, speech processing, and broadband wireless communication.