MERGED INVERSE QUANTIZATION AND IDCT FOR OPTIMIZED DECODER IMPLEMENTATION

Jin Li[1]*, Moncef Gabbouj*[1]*, Jarmo Takala*[1] *and Hexin Chen*[2]

Faculty of Computing and Electrical Engineering, Tampere University of Technology, Finland[1]
School of Communication Engineering, Jilin University, Changchun, China[2]
FirstName.LastName@tut.fi[1] chx@jlu.edu.cn[2]

ABSTRACT

This paper proposes an efficient technique to reduce the complexity of implementation for inverse discrete cosine transform (IDCT)-based video decoders. The proposed method merges the inverse quantization and IDCT into a single procedure, which is referred to as inverse quantized IDCT (IQIDCT), such that the decoded coefficients do not need to be inverse quantized prior to the inverse transform. Thus, the computations related to inverse quantization are omitted. Since the cosine transform basis can be constructed before decoding, the proposed IQIDCT effectively replaces the need for computing power with little additional memory. The performance of the proposed algorithm is evaluated by comparing it with the original XVID decoder which uses the traditional separate inverse quantization and IDCT method. Experiments show the efficiency of the proposed method in reducing the overall decoding complexity. Moreover, it does not result in any video quality degradation. The proposed method is particularly suitable for low-power processors in multimedia systems.

1. INTRODUCTION

Traditionally, the objective in video coding has been high compression efficiency, which is usually achieved at the expense of increasing computational complexity. As the newest video coding standard, H.264/AVC [1] significantly outperforms others in terms of coding efficiency. However, the complexity is greatly increased at both encoder and decoder. As real-time video applications have become commonplace for mobile smart phones, PDAs and personal multimedia systems, the software video codec on low-power processors still requires significant amount of computations as well as battery power. Thus, there is great significance and interest in reducing the computations for efficient implementation of video coding.

Even though decoder has less complexity than encoder in video standards, the importance of low decoding complexity is equal to or even more than low encoding complexity. The reason is that, in many applications such as DVD players, digital TV receivers *etc*, the end-user equipment has only the decoder implemented and the decoder block is the only codec related functional block adding complexity to the system. Even in a system where encoder and decoder co-exist, the decoding complexity could become very important if the encoder and the decoder are running on different hardware platforms.

Many algorithms have been proposed to reduce the decoding complexity for optimized decoder implementation. A. Navarro *et al.* [2] propose a fast integer IDCT to speed up the decoding by using more efficient transform structure. In 2006, an algorithm is proposed by K. Ugur *et al.* [3] to relieve the computational load in the decoder of H.264. The method is able to generate decoder-friendly bit stream at the encoder and reduce the decoding complexity at the cost of negligible video degradation. Recently, S. W. Lee *et al.* [4] propose a complexity reduction model for H.264/AVC decoder. The H.264 encoder integrated with the proposed model selects proper coding mode to minimize the distortion while satisfying the decoding complexity constraints. In addition, many efforts have been done for the optimized implementation of video decoding on various processors [5], [6]. All the methods can reduce the decoding complexity and optimize the implementation of decoders. However, since they are utilizing the traditional separate inverse quantization and IDCT method, the calculations for inverse quantization are not reduced.

In what follows we describe an efficient method to reduce the decoding complexity. The proposed technique merges the inverse quantization and IDCT into a single procedure. Thus, the computations related to inverse quantization are skipped. The proposed method can be directly applied to other existing methods such as [2]-[6] and further optimize the implementation of the decoders.

The rest of this paper is organized as follows. Section 2 proposes the IQIDCT algorithm and the implementation. The experimental results are presented and discussed in Section 3. Finally, Section 4 concludes this paper.

2. PROPOSED IMPLEMENTATION OF INVERSE QUANTIZATION AND IDCT

In this section, we describe in detail the proposed IQIDCT algorithm and the implementation of 2-D IDCT-based video decoding.

2.1 Combination of Inverse quantization and IDCT

In this paper, we mainly consider the 8×8 2-D IDCT which is widely used in the H.263 and MPEG-4 decoder. If we define $F^q(u,v) = [\boldsymbol{F^q}]_{u,v}, 0 \le u, v \le 7$, as the quantized DCT coefficient at the decoder that are retrieved from the decoded bit streams, the inverse quantized DCT coefficient $F(u,v)$ is computed by

$$F(u,v) = F^q(u,v) \times Q(u,v) \tag{1}$$

where $Q(u,v)$ is the inverse quantization.

Following the operation of inverse quantization, the 2-D IDCT transform is defined as [7]

$$f(x,y) = \frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7} c(u)c(v)F(u,v)\cos\frac{(2x+1)u\pi}{16}\cos\frac{(2y+1)v\pi}{16} \tag{2}$$

where $c(u), c(v) = 1/\sqrt{2}$, for $u, v = 0$, and $c(u)$, $c(v) = 1$, otherwise.

As 2-D IDCT is a linear and separate transform, the butterfly row-column structure is usually used in video decoders for efficient implementation. At first, we consider the row transform of eight-point 1-D IDCT which is defined as

$$f(x,v) = \frac{1}{2}\sum_{u=0}^{7} c(u)F(u,v)\cos\frac{(2x+1)u\pi}{16} \tag{3}$$

The corresponding matrix representation is

$$\boldsymbol{f(v)} = \boldsymbol{CF(v)} \tag{4}$$

where $\boldsymbol{f(v)}$ is an eight-point 1-D IDCT coefficient vector at the $\boldsymbol{v}$th column, $\boldsymbol{F(v)}$ is the input vector of the 2-D DCT coefficients and $\boldsymbol{C}$ denotes an 8×8 cosine transform basis where seven coefficients fully describes the matrix, shown as

$$C = \begin{bmatrix} C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 & C_4 \\ C_1 & C_3 & C_5 & C_7 & -C_7 & -C_5 & -C_3 & -C_1 \\ C_2 & C_6 & -C_6 & -C_2 & -C_2 & -C_6 & C_6 & C_2 \\ C_3 & -C_7 & -C_1 & -C_5 & C_5 & C_1 & C_7 & -C_3 \\ C_4 & -C_4 & -C_4 & C_4 & C_4 & -C_4 & -C_4 & C_4 \\ C_5 & -C_1 & C_7 & C_3 & -C_3 & -C_7 & C_1 & -C_5 \\ C_6 & -C_2 & C_2 & -C_6 & -C_6 & C_2 & -C_2 & C_6 \\ C_7 & -C_5 & C_3 & -C_1 & C_1 & -C_3 & C_5 & -C_7 \end{bmatrix}$$

where $C_n = \frac{1}{2}\cos(n\pi/16)$, and $n = 1, \ldots, 7$.

Since the DCT coefficient $F(u,v)$ are calculated from the quantization $Q(u,v)$ and the quantized coefficient $F^q(u,v)$ as (1), the row inverse cosine transform of $F(u,v)$ can be directly represented as

$$f(x,v) = \frac{1}{2}\sum_{u=0}^{7} c_u(v)F^q(u,v) \tag{5}$$

where

$$c_u(v) = Q(u,v)c(u)cos\frac{(2x+1)u\pi}{16}$$

As shown in (4), the inverse quantization $\boldsymbol{Q}$ is embedded into the cosine transform matrix $\boldsymbol{C}$**.** Thus**,** the operations for the inverse quantization prior to the IDCT are not required. The row one-dimensional IDCT in (5) is represented in matrix form as

$$\boldsymbol{f(v)} = \boldsymbol{C^q(v)F^q(v)} \tag{6}$$

where $\boldsymbol{C^q(v)}$ is the transform basis matrix at the $\boldsymbol{v}$th column embedded by the inverse quantization $\boldsymbol{Q(v)}$ shown as

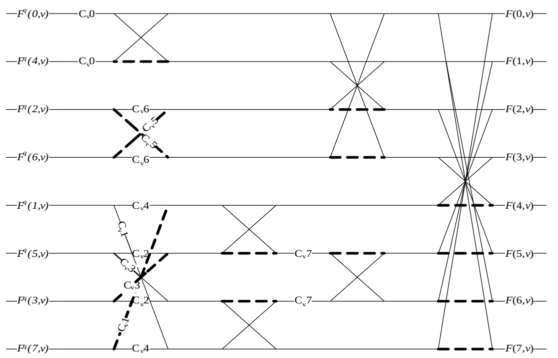
$$C^q(v) = \begin{bmatrix} C^q_{4,0}(v) & C^q_{4,0}(v) & C^q_{4,0}(v) & C^q_{4,0}(v) & C^q_{4,0}(v) & C^q_{4,0}(v) & C^q_{4,0}(v) & C^q_{4,0}(v) \\ C^q_{1,1}(v) & C^q_{3,1}(v) & C^q_{5,1}(v) & C^q_{7,1}(v) & -C^q_{7,1}(v) & -C^q_{5,1}(v) & -C^q_{3,1}(v) & -C^q_{1,1}(v) \\ C^q_{2,2}(v) & C^q_{6,2}(v) & -C^q_{6,2}(v) & -C^q_{2,2}(v) & -C^q_{2,2}(v) & -C^q_{6,2}(v) & C^q_{6,2}(v) & C^q_{2,2}(v) \\ C^q_{3,3}(v) & -C^q_{7,3}(v) & -C^q_{1,3}(v) & -C^q_{5,3}(v) & C^q_{5,3}(v) & C^q_{1,3}(v) & C^q_{7,3}(v) & -C^q_{3,3}(v) \\ C^q_{4,4}(v) & -C^q_{4,4}(v) & -C^q_{4,4}(v) & C^q_{4,4}(v) & C^q_{4,4}(v) & -C^q_{4,4}(v) & -C^q_{4,4}(v) & C^q_{4,4}(v) \\ C^q_{5,5}(v) & -C^q_{1,5}(v) & C^q_{7,5}(v) & C^q_{3,5}(v) & -C^q_{3,5}(v) & -C^q_{7,5}(v) & C^q_{1,5}(v) & -C^q_{5,5}(v) \\ C^q_{6.6}(v) & -C^q_{2,6}(v) & C^q_{2,6}(v) & -C^q_{6,6}(v) & -C^q_{6,6}(v) & C^q_{2,6}(v) & -C^q_{2,6}(v) & C^q_{6,6}(v) \\ C^q_{7,7}(v) & -C^q_{5,7}(v) & C^q_{3,7}(v) & -C^q_{1,7}(v) & C^q_{1,7}(v) & -C^q_{3,7}(v) & C^q_{5,7}(v) & -C^q_{7,7}(v) \end{bmatrix}$$

where $C^q_{n,x}(v) = \frac{1}{2}Q(x,v)cos\,(n\pi/16)$, $n = 1, 2, \ldots, 7$ and $0 \le x, v \le 7$.

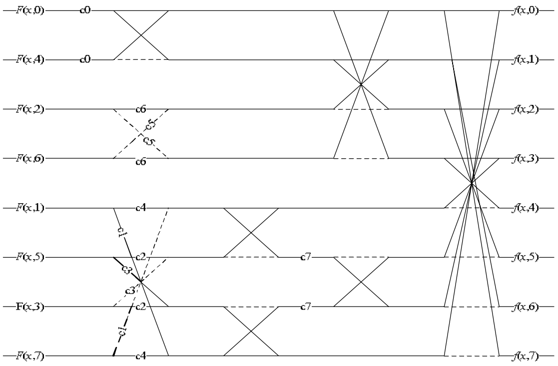
Compared to the transform matrix $\boldsymbol{C}$, the modified matrix $\boldsymbol{C^q(v)}$ merges inverse quantization into the row inverse transform and comprise maximum 43 different components. In addition, since the quantization $\boldsymbol{Q}$ is not necessary to be a uniform matrix, the transform basis matrix $\boldsymbol{C^q(v)}$ is usually different for different columns.

Following is the column inverse transform for 2-D IDCT. Since the proposed IQIDCT merges the inverse quantization and the row inverse transform into a single procedure, the 2-D IDCT coefficients $\boldsymbol{f}$, i.g. the reconstructed pixels, can be computed form the cosine transform basis $\boldsymbol{C}$ and the 1-D IDCT coefficients $\boldsymbol{f(v)}$ as

$$\boldsymbol{f} = \boldsymbol{f'C^T} \tag{7}$$



(a)



(b)

Fig.1 (a) Signal flow graph of row transform used in IQIDCT, where $\boldsymbol{F^q}$ is the 2-D DCT coefficient prior to inverse quantization, $\boldsymbol{C_v}$ is the transform basis at the $\boldsymbol{v}$th column as (6) and $\boldsymbol{F}$ is the 1-D IDCT coefficients. (b) Chen's structure, where $\boldsymbol{f}$ are the 2-D IDCT coefficients and $\boldsymbol{C}$ is the transform basis as (4).

where $\boldsymbol{C^T}$ is the transpose matrix of $\boldsymbol{C}$ in (4) and $\boldsymbol{f'}$ denotes an 8×8 matrix composed of the eight $\boldsymbol{f}(\boldsymbol{v})$ vectors in (6).

Therefore, using the proposed IQIDCT algorithm the 2-D IDCT coefficients are directly calculated from the quantized DCT coefficients and a set of new transform basis matrix without the inverse quantization. As for the calculations of 2-D IDCT, which is usually implemented by the butterfly row-column transform method, our proposed algorithm only modifies the cosine coefficients of the row inverse transform by interpolating the components of inverse quantization. At the second transformation stage, i.e. the column 1-D IDCT at our work, the transform is performed in the normal way as (7).

Based on the above analysis, the proposed IQIDCT is a nonlinear transform by embedding the inverse quantization into the 1-D IDCT transform, i.e. the row transformation stage in our work. The main idea is to pre-compute and store a set of coefficients for each quantizer used in the decoder. This effectively replaces the need for computing power with little additional memory in hardware implementation. In the following we discuss in detail the implementation as well as the required memory for both uniform and non-uniform quantization, which is widely used for inter coding and intra coding.

2.2 Implementation of proposed method in 2-D DCT

Practical systems in video decoders do not compute the 2-D IDCT coefficients directly using (2). Fast implementations exploiting the particular structure of the cosine transform matrix $\boldsymbol{C}$ have been proposed that require significantly fewer computations. The proposed modifications to the IDCT can be applied to any of the fast algorithms. In this work, we use Chen's algorithm [8] to compute the proposed algorithm. Utilizing the separate property of the 2-D IDCT, the first transformation stage, i.e. the row inverse transform in our work, is modified by the proposed IQIDCT. The second transformation stage, i.e. the column transform, still follows Chen's structure. The flow graph for the proposed IQIDCT is sketched in Fig. 1.

Since the cosine transform matrix $\boldsymbol{C}$ are modified by embedding the inverse quantization into the row 1-D DCT, additional memory is required to store a set of new basis coefficients $\boldsymbol{C^q}$ that are pre-computed before the decoding process. For non-uniform quantization widely used in intra coding, the maximum number of modified coefficients to be stored at the memory is $64 \times 8 = 512$ for the 8×8 IDCT-based decoder. For the uniform quantization in inter coding, since the quantization values are invariable, the maximum required number of additional coefficients is 64. Compared to the computational reduction obtained by the proposed technique, the increasing in memory is negligible.

The implementation of the proposed IQIDCT algorithm is summarized as follows

1) The new transform basis $\boldsymbol{C^q}$ at (6) are pre-computed by embedding the inverse quantization into the transform basis matrix $\boldsymbol{C}$ and stored in the memory.
2) The row transform is applied with the proposed IQIDCT shown in Fig.1 (a): using the modified transform basis $\boldsymbol{C^q}$ to compute the intermediate 1-D IDCT coefficients.
3) The column transform is performed using the normal IDCT algorithm with the signal flow graph in Fig.1 (b).

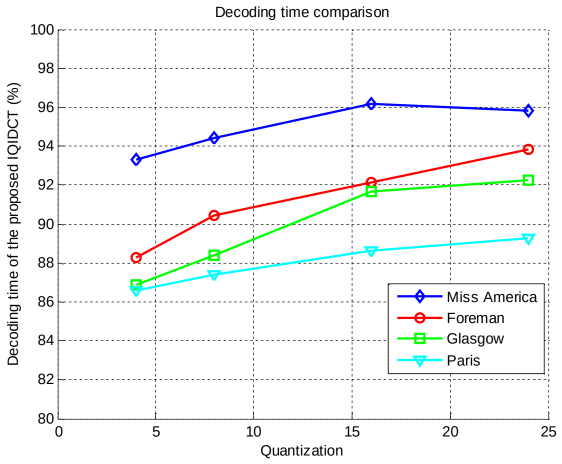


Fig.2. Comparison of decoding time of the proposed IQIDCT and the XVID decoder for Miss America, Glasgow, Foreman and Paris

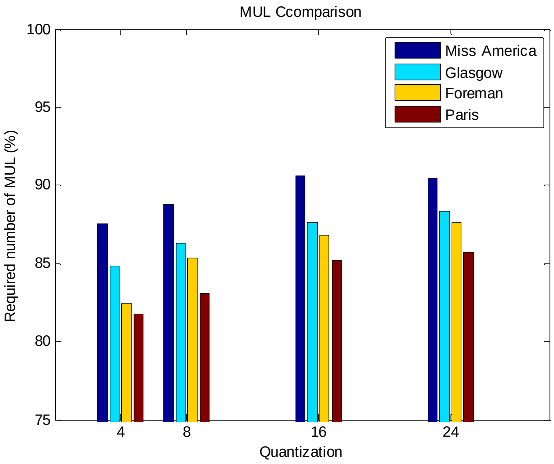


Fig.3. Complexity reduction in terms of MUL between the proposed IQIDCT and the XVID decoder for Miss America, Glasgow, Foreman and Paris

3. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithm, a series of experiments were carried out against the MPEG-4 based XVID decoder [9]. In the experiments, we apply the proposed method to both intra coding and inter coding, thus the inverse quantization in the decoder does not need to be computed. Totally nine new cosine transform matrix $\boldsymbol{C^q}$ is pre-computed and stored in the memory. Four video sequences in QCIF format (Glasgow, Paris, Miss America and Foreman) are tested. All the simulations are running on a PC with Intel Pentium 2.0G and 768Mbytes of RAM. Different quantization parameter is used to examine the performance at different bit rates.

Firstly, we will study the real-time performance of the proposed IQIDCT. The comparison of the decoding time between the proposed method and the XVID decoder is shown in Fig.2. In the figure, the decoding time for the proposed IQIDCT is defined as

$$\nabla T = \frac{T}{T_{org}} \times 100\% \qquad (8)$$

where T_{org} and T are the entire decoding time of the XVID decoder and the test model.

From Fig.2, it is shown that the proposed method achieves better real-time performance. This validates that the proposed IQIDCT method can reduce the computational complexity of the decoder and is superior to the traditional separate inverse quantization and transform method. In particular, based on the experiments the proposed IQIDCT is more efficient at high bit rates. The reason is that, since more nonzero quantized DCT coefficients are retrieved from the decoded bit streams at high bit rates, the number of omitted operations for inverse quantization is more than that at low bit rate. Take Paris for example, when it is coded at $Q_p = 4$, the running time is reduced to 88.58% of the original XVID decoder, while the entire decoding time increase to 91.27% when Q_p is increasing to 24.

As an important factor for the implementation of decoder, the complexity reduction in terms of multiplication (MUL) is also taken into consideration. Compared to the traditional separate method, the proposed IQIDCT does not require the MUL calculation for inverse quantization, thus a lot of MUL operations are saved. The overall required number of MUL in the test model is compared with the XVID decoder as shown in Fig.3. Experiments show that the proposed IQIDCT reduces the number of MUL by 7.53-17.26%. Reduction in MUL operations benefits a lot of low-power processors in portable devices such as mobile phones, since MUL usually consumes more power than other operations.

The objective video quality in terms of Peak signal-to-noise ratio (PSNR) against bit rates is compared between the proposed method and the XVID decoder. No video quality degradation is observed based on the experimental results. This further validates that the proposed IQIDCT only omits the calculations for the traditional inverse quantization while does not deteriorate the computational precision in the decoding process.

Overall, the proposed model can reduce the required computations of IDCT and inverse quantization and speed up the decoding process. Compared to the reference encoder, the proposed method is able to further reduce the multiplications for inverse transform and quantization and thus, has better performance for low-power processors. Moreover, the experiments show that the proposed method does not cause to any video quality degradation.

4. CONCLUSIONS

This paper proposes an efficient technique that merges the inverse quantization and IDCT into a single procedure, such that the decoded coefficients do not need to be inversely quantized. Experiments show that the proposed method can reduce the decoding complexity and optimize the real-time implementation than the XVID decoder. This effectively replaces the need for computing power with little additional memory for implementation. In addition, it is particularly suitable for low-power processors in portable devices where MUL consumes more power than others. Experimental results also show that the proposed IQIDCT method does not degrade the coding efficiency.

5. ACKNOWLEDGEMENT

This work was supported by the Academy of Finland, project No. 213462 (Finnish Centre of Excellence Program (2006 - 2011), the Academy of Finland Grant 117065 and by the Chinese Science & Technology Ministry Grant 2005DFA10300. The authors thank Nokia Foundation for the support of research work.

6. REFERENCES

[1] J. Smith, and H. Kalva, “The H.264 video coding standards,” *IEEE Trans. Multimedia*, Vol.13, No.4, pp.86-90, Dec. 2006.

[2] A. Navarro, A. Silva, and J. Tavares, “MPEG-4 codec performance using a fast integer IDCT,” *IEEE Int. Symp. on Consumer Electronics*, pp.1-5, St.Petersburg, Russia, 2006.

[3] K. Ugur, J. Lainema, A. Hallapuro, and M. Gabbouj, “Generating H.264/AVC compliant bitstreams for lightweight decoding operation suitable for mobile multimedia systems,” *IEEE ICASSP*, Vol. II, pp.33-36, Toulouse, France, 2006.

[4] S.W. Lee, and C. Kuo, “Complexity modeling of H.264/AVC CAVLC/UVLC entropy decoders,” *IEEE ISCAS*, pp.1616-1619, Seattle, USA, 2008.

[5] H. Liu, R. Xu, and Z. Liu, “Implementation of H.264 on TMS320DM642,” *IEEE ICESS*, pp.605-609, Chengdu, China, 2008.

[6] V. Lappalainen, A. Hallapuro, T. Hamalainen, “Complexity of optimized H.26L video decoder implementation,” *IEEE Trans. Circuits and Syst. for Video Tech.*, Vol.13, No.7, pp.717-725, July, 2003.

[7] V. Bhaskaran and K. Konstantinides, “Image and video compression standards: Algorithms and Architecture,” Norwell, MA: Kluwer, 1995.

[8] K. Rao, and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*. New York: Academic, 1990.

[9] [online] http://www.xvid.org/