A PRACTICAL SOLUTION TO TRANSFORM-DOMAIN ROUNDING

Chen Chen, Ping-Hao Wu, and Homer Chen

Graduate Institute of Communication Engineering, National Taiwan University
Taipei, Taiwan 10617, R.O.C.
email: {r92942081, b89901043}@ntu.edu.tw and homer@cc.ee.ntu.edu.tw

ABSTRACT

In many multimedia applications, manipulation of video signals is preferably performed in the transform domain. Most of such operations are derived from their counterparts in the pixel domain. However, the rounding operation can not be obtained this way because it is a nonlinear operation. Without a proper treatment, it may become a major source of errors for multimedia systems, causing a mismatch between the encoder and the decoder. This problem seems to have been largely ignored. In this paper, we propose a practical solution to this problem by a thorough examination of the mathematical property of the rounding error in the context of transform-domain video transcoding and demonstrate its performance.

1. INTRODUCTION

As the technologies of digital signal processing and VLSI advance, multimedia services have become a focus of interest. Usually, video signals are compressed before they are stored or transmitted over networks. In many applications, such as transcoding, manipulation of video signals directly on the compressed bit streams is more desirable than the approach where the compressed videos are fully decoded and then manipulated, because of the computational cost issue [1].

Video coding standards normally apply discrete cosine transform (DCT) to the video signal to achieve energy compaction. Video signals in the pixel domain are converted to the transform domain. For image operations such as motion compensation and down sampling, their counterparts in the transform domain can be easily derived [2] by making use of the orthogonality of DCT and the linearity of these operations [2].

The rounding operation is commonly performed in multimedia systems. Take half-pel motion compensation as an example. The generation of new pixels at half-pel locations requires a rounding operation for better approximation of the pixel values. However, rounding is a nonlinear operation; so its corresponding transform-domain operation does not exist. To solve the problem, Shanableh and Ghanbari proposed an approach that adds a constant value of 4 to the DC component of an 8x8 DCT transform [3], while most other transform-domain video transcoding schemes (for example, [1], [4]) choose to skip the rounding operation completely. All such approaches suffer from video quality degradation due to the accumulation of drift errors. Without a proper treatment of this problem, the output video quality of transform-domain transcoding will not be comparable to that of the pixel-domain transcoding.

In this paper, we present an approximate solution to this important problem by a thorough examination of the mathematical property of the rounding error. The approach we develop is able to minimize the drift error measured between the pixel-domain transcoding and the transform-domain tanscoding.

This paper is organized as follows. In Section 2, we describe the problem of rounding in the transform domain and review previous approaches to this problem. The basic idea of the approximate solution is described in Section 3. The details of the proposed algorithm to deal with the rounding problem are given in Section 4. Then, the simulation results are described in Section 5, followed by a conclusion in Section 6.

2. PROBLEM DESCRIPTION

For an image block in the pixel domain, the rounding operation (addition of a constant value 0.5, followed by a truncation) can be easily performed by checking if the fractional part of each pixel value is greater than or equal to 0.5. This is equivalent to adding a block of 0.5 to the image block and truncating the resulting image block. When the block size is 8, adding a block of 0.5 in the pixel domain is equivalent to adding 4 to the DC coefficient of the image block in the DCT (or transform) domain [3]. To complete the rounding operation in the transform domain, another operation equivalent to the truncation operation in the pixel domain has to be performed. However, the truncation is a nonlinear operation. Its equivalent operation in the transform domain is undefined. As a result, merely adding 4 to the DC coefficient without applying a mathematically equivalent operation of truncation would inevitably introduce errors to the results.

3. THE BASIC IDEA

Fig. 1 shows the flow of a typical averaging operation. Suppose Input 1 to Input n and Output are all positive integers. Our idea behind the proposed algorithm is based on the observation that the possible fractional value after division is a finite set, and that the size of the set is determined by the divisor. For example, if the divisor is 4, then the fractional value after division is a member of the set
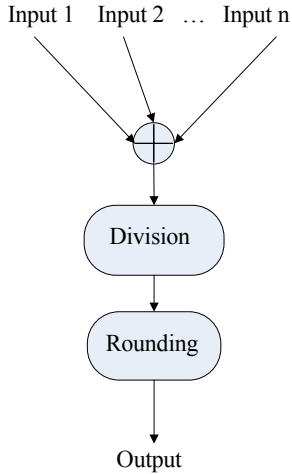
Fig. 1. Illustration of a typical averaging operation.

{0, 0.25, 0.5, 0.75}, and the value difference before and after rounding is a member of the set {0, -0.25, 0.5, 0.25}. Suppose that it is equally likely for the value before division to be odd or even number, then the value difference before and after rounding can be modelled by a discrete uniform distribution. Based on the model, we describe an approach to approximating the rounding operation in the next section.

4. PROPOSED ALGORITHM

Considering the case described in section 3, the possible value difference before and after rounding is 0, -0.25, 0.5, or 0.25 with equal probability. Let R be the random variable representing the value difference before and after rounding, then R is uniformly distributed with the probability mass function

$$p_R(r) = \frac{1}{4}\delta(r+0.25) + \frac{1}{4}\delta(r) + \frac{1}{4}\delta(r-0.25) + \frac{1}{4}\delta(r-0.5),$$

(1)

where $\delta(\cdot)$ is an impulse function. Now we want to find a value s, such that the expectation of $(R-s)^2$, which is denoted by $E[(R-s)^2]$, is minimum. By simple calculation, it can be found that s is equal to the expectation of R (i.e. E[R]). Therefore, the value difference before and after rounding can be approximated by E[R]. Then, the rounded value is obtained by adding E[R] to the original value.

Consider a general block-wise data in Fig. 2. Block X is composed of the fractional values before rounding, block Y is the result of rounding block X, and block D is the difference between X and Y. The block sizes of X, Y, and D are denoted by n. Note that, as we assumed, the elements of block D are all uniformly distributed. To minimize the expected mean square error, take the expectation of each element as the estimate. In the transform domain, the estimates are written as

$$T\begin{pmatrix} E[d_{1,1}] & \cdots & E[d_{1,n}] \\ \vdots & \ddots & \vdots \\ E[d_{n,1}] & \cdots & E[d_{n,n}] \end{pmatrix},$$
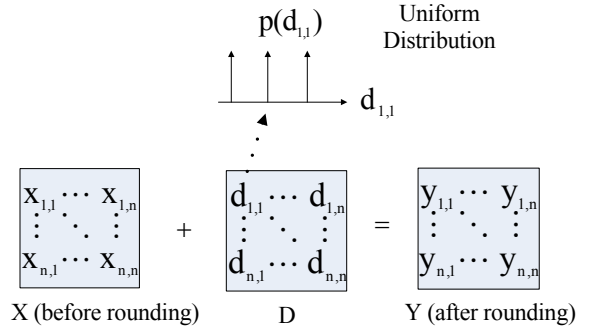
(2)

Fig. 2. The probability model of the value difference before and after rounding.

where $T(\cdot)$ denotes the DCT transform. Then, the transform-domain rounding operation can be approximated by adding to the original block.

Using this approximation in the transform domain, the squared error is minimized. Moreover, if the divisor is known in advance, assuming a uniform distribution, the expectation of each element can be pre-computed, and this makes the proposed approach easy to implement.

5. SIMULATION RESULTS

In this section, we report the result of an experiment in which we compare the performance of the proposed approach to two existing approaches. One of these two approaches, called *Without Truncation,* adds a block of 0.5 to original block but does not apply the truncation operation. The other approach, *Without Rounding,* skips the rounding process completely.

Fig. 3 shows our first experiment, where "a" is equal to 0.5, and "b" is equal to 0.125. Three different operations apply to a 4x4 block X. Y is obtained by rounding X, and serves as the benchmark representing the pixel domain result. Y_t and Y_p are obtained by using *Without Truncation* approach and the proposed approach, respectively.

In the simulations, X is randomly generated, where each element in X is uniformly distributed in the set {0, 0.25, 0.5, 0.75}. Y_t and Y_p are compared with Y, and the performance of each approach is measured in terms of mean-square error
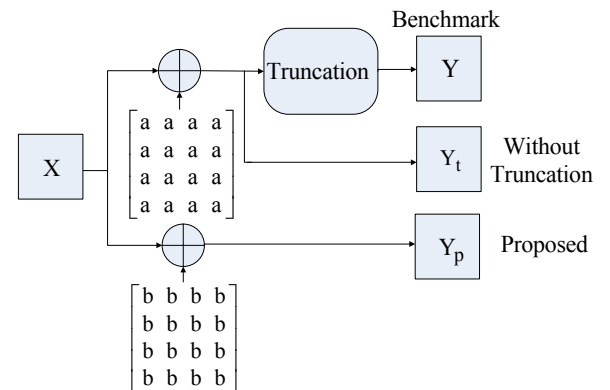

Fig. 3. The block diagram of the first experiment.

(MSE) in the pixel domain and the transform domain. We make 10000 runs and calculate the average MSE.

Table 1 shows the average MSE of each approach. It can be seen that the error of the proposed method is less than that of other approaches that is expected since adding a block of 0.125 to the original block minimizes the error when the probability assumption is held. In addition, the value difference between each element in Y_p and Y_t is 0.375, which is more than the difference of each element between Y_p and X. Therefore, the approach of *Without Truncation* induces more error than the approach of *Without Rounding*.

In the second experiment, we apply the proposed method to the transform-domain motion compensation in an MPEG-2 decoder. The transform-domain motion compensation with half-pixel accuracy is performed by averaging either two or four motion compensated blocks. When both components of the motion vector have half-pixel accuracy, the divisor is 4. According to the proposed algorithm, the rounding operation is approximated by adding 1 to the DC coefficient of the image block in the transform domain. When only one component of the motion vector, horizontal or vertical component, has half-pixel accuracy, the divisor is 2. Then the rounding operation is approximated by adding 2 to the DC coefficient of the image block in the transform domain.

In the simulations, two 100-frame CIF size (352x288) test sequences, "Foreman" and "Tempete," are coded with GOP (Group of Pictures) structure (100, 1) (i.e., IPPP…). The bitrate of the MPEG-2 bitstream is 4Mbits/s.

Fig. 4 and 5 are the PSNR plots of the decoded video, which show the impact on the quality when using different approaches. We observe that the drift error is the least for the proposed approach. Several frames of decoded *Foreman* sequences are shown in Fig. 6 to 8 for subjective comparison. As we can see, the motion compensation process without performing rounding operation accumulates errors, which in turn cause the picture to turn green, as shown in Fig. 6 (b), 7 (b), and 8 (b). When the DC value is added by an inappropriate value, the picture turns red, as Fig. 6 (c), 7 (c), and 8 (c).

6. CONCLUSTION

In this paper, we have investigated the drift between pixel-domain transcoding and transform-domain transcoding due to rounding and proposed a mathematical model for the fractional value before rounding. Based on the model, we develop an approach, adding the appropriate values to the transform coefficients, to approximate the rounding operation. The proposed approach minimizes the expected mean square error and is easy to implement. We have also demonstrated the performance of the approach when applied to an MPEG-2 decoder and compared it with competing approaches. The superior performance shows that the proposed method is a practical solution.

Table 1. The comparison of average mean square errors.

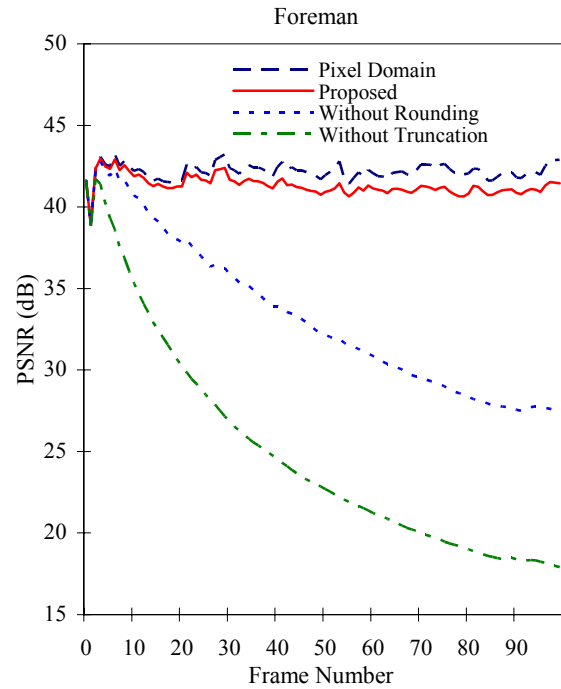|	Proposed	Without Rounding	Without Truncation
MSE	0.0781	0.1254	0.2184

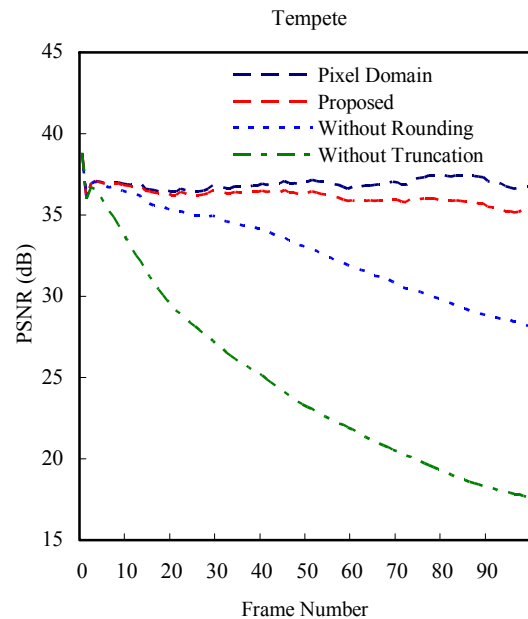Fig. 4. PSNR comparison for the Foreman sequence.

Fig. 5. PSNR comparison for Tempete sequence.

REFERENCES

[1] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 84-97, Jan. 2005.

[2] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Select. Areas Commun.*, vol. 13, pp. 1-11, Jan. 1995.

[3] T. Shanableh and M. Ghanbari, "Hybrid DCT/pixel domain architecture for heterogeneous video transcoding," *Signal Processing: Image Commun.*, vol. 18, no. 8, pp. 601-620, Sep. 2003.

[4] P. A. A. Assuncao and M. Ghanabari, "A frequency-domain video transcoder for dynamic bit-rate reduction of MPEG-2 bit streams," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, pp. 953-967, Dec. 1998.

[5] N. Merhav and V. Bhaskaran, "Fast algorithm for DCT-domain image down-sampling and for inverse motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp.468-476, Jun. 1997.

[6] A. Vetro, C. Christopulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing mag.*, vol. 20, no.2, pp. 18-29, Mar. 2003.

[7] J. Song and B.-I. Yeo, "A fast algorithm for DCT-domain inverse motion compensation based on shared information in a macroblock," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 5, pp. 767-775, Aug. 2000.

[8] Y.-R. Lee, C.-W. Lin, and Y.-W. Chen, "Computation reduction in cascaded DCT-domain video downscaling transcoding," in *Proc. IEEE Int. Symp. Circuits and Systems*, 2003, pp. 860-863.
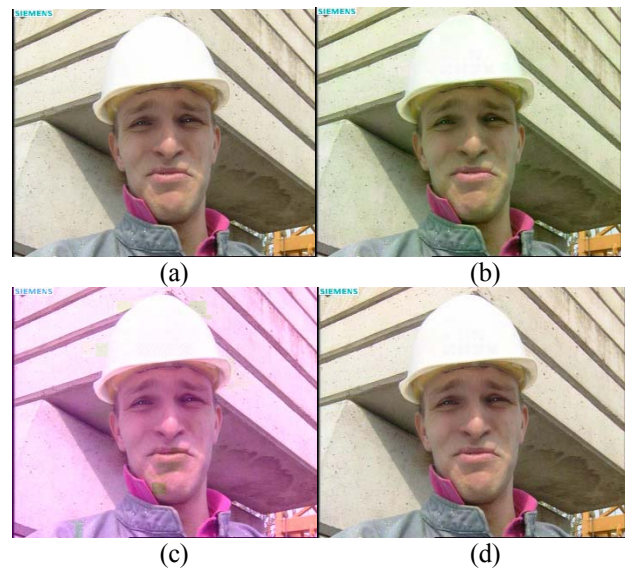
Fig. 7. The 50th frame of the Foreman sequence. The decoded images obtained from (a) the pixel domain approach, (b) the transform domain approach without rounding, (c) the transform domain approach without truncation, and (d) the transform domain approach with proposed algorithm.

Fig. 8. The 100th frame of the Foreman sequence. The decoded images obtained from (a) the pixel domain approach, (b) the transform domain approach without rounding, (c) the transform domain approach without truncation, and (d) the transform domain approach with proposed algorithm.
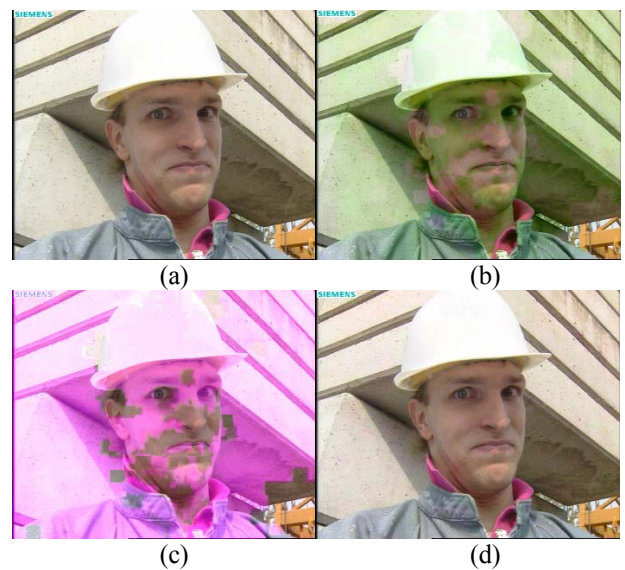

Fig. 6. The 10th frame of the Foreman sequence. The decoded images obtained from (a) the pixel domain approach, (b) the transform domain approach without rounding, (c) the transform domain approach without truncation, and (d) the transform domain approach with proposed algorithm.