

Rounding Mismatch Between Spatial-Domain and Transform-Domain Video Codecs

Ping-Hao Wu, Chen Chen, and Homer H. Chen, *Fellow, IEEE*

Abstract—The mismatch between a spatial-domain encoder and a transform-domain decoder due to rounding can cause serious video quality degradation after a series of inter-coded frames. The mismatch is rooted on the fact that rounding is a nonlinear operation, and hence its equivalent in the transform domain does not exist. We analyze the mathematical property of the rounding operation and propose a practical solution that provides an optimal approximation of the rounding operation in the transform-domain. The proposed solution is able to reduce the mismatch and preserve image quality. Experimental results are shown to demonstrate the effectiveness of the solution.

Index Terms—Discrete cosine transform (DCT), motion compensation, rounding, transcoding, transform domain.

I. INTRODUCTION

IN MANY video applications, real-time manipulation of the video signal in the transform domain is desired because of its computational advantage [7]–[9], [11]–[16], [18]. For example, a decoder which performs the motion compensation (MC) in the discrete cosine transform (DCT) domain can be much faster than a spatial-domain decoder (SD) [9], [14], [15]. To manipulate the video signals in the transform domain, the compressed bitstream is first decoded to DCT coefficients through a variable length decoder, followed by a DCT-domain MC (MC-DCT) operation. Then image manipulations are performed on the resulting DCT coefficients.

Most operations performed by a video coding system are linear [1]–[6]. Owing to the orthogonality of DCT [9], such operations can be directly converted to the transform domain. Results obtained by these operations in the transform domain are equivalent to the results obtained in the spatial domain. However, the rounding operation, which is often used in subpixel MC, does not have a counterpart in the transform domain, resulting in a mismatch between a spatial-domain encoder (SE) and a transform-domain decoder (TD). As in [13] and [16], the term “transform domain decoder” is used to refer to a decoder that performs MC in the transform domain. Also, we only discuss the rounding mismatch in this paper. Other types of mismatches (such as IDCT mismatch) are not considered.

Manuscript received September 12, 2005; revised April 18, 2006. This work was supported in part by the National Science Council of Taiwan under Contracts NSC 93-2752-E-002-006-PAE, NSC93-2213-1-002-137, and NSC 93-2219-1-002-031. This paper was recommended by Associate Editor K.-H. Tzou.

P.-H. Wu and C. Chen are with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C.

H. H. Chen is with the Department of Electrical Engineering, Graduate Institute of Communication Engineering, and Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: homer@cc.ee.ntu.edu.tw).

Digital Object Identifier 10.1109/TCSVT.2006.881855

Due to the recursive nature of motion-compensated video coding, any mismatch error could propagate through a series of inter-coded frames and severely degrade the video quality. To ensure the interoperability between an encoder and a decoder supplied by two different vendors, the mismatch between spatial-domain and transform-domain coders has been studied. In [13], it is shown that the mismatch only occurs when the decoder’s MC is performed in a domain different from the encoder’s. Two criteria are established for matching spatial-domain and transform-domain codecs. The first criterion is the distributive property of inverse DCT (IDCT), and the second is the commutative property of IDCT and MC.

In [16], the mismatch problem is further analyzed, and a precision lifting method is proposed. However, to ensure identical coding states at the encoder and the decoder sides, the precision lifting method has to be applied at both sides, which makes this approach unattractive since, in practice, the decoder does not know if the encoder adopts this approach.

The mismatch due to rounding is rooted on the fact that the rounding is a nonlinear operation. Some transform-domain video transcoding schemes [10], [12] choose to ignore the rounding operation and, as a result, suffer from video quality degradation and, worse, accumulation of drift errors. Shanableh and Ghanbari [11] proposed an approach to deal with the rounding problem. Their work is mostly related to ours.

In this paper, an approach that has sound theoretical basis and significant performance enhancement is presented. We examine the mathematical property of the rounding operation and show that an optimal approximation of the rounding operation in the transform domain can be achieved by minimizing the mean-square error (MSE) between a TD and a conventional SE, and that the transform-domain rounding becomes a bias compensation operation. The approach can effectively reduce the drift error and preserve image quality.

This paper is organized as follows. In Section II, we analyze the mismatch between a SE and a TD and describe the problem associated with the rounding operation in the transform domain. Section III gives a review of previous approaches. The basic idea and the details of the proposed algorithm are described in Section IV. In Section V, the experiment results are presented, followed by a conclusion in Section VI.

II. ANALYSIS OF THE MISMATCH BETWEEN SD AND TD

A mismatch between the SE and a TD occurs when the decoder performs MC in a domain different from that of the encoder.

Fig. 1 depicts a SE. A SD and a TD are shown in Figs. 2 and 3. The mismatch problem arises when the MC in the decoder is not equivalent to that in the encoder. The conventional SE can be described by iterative functions.

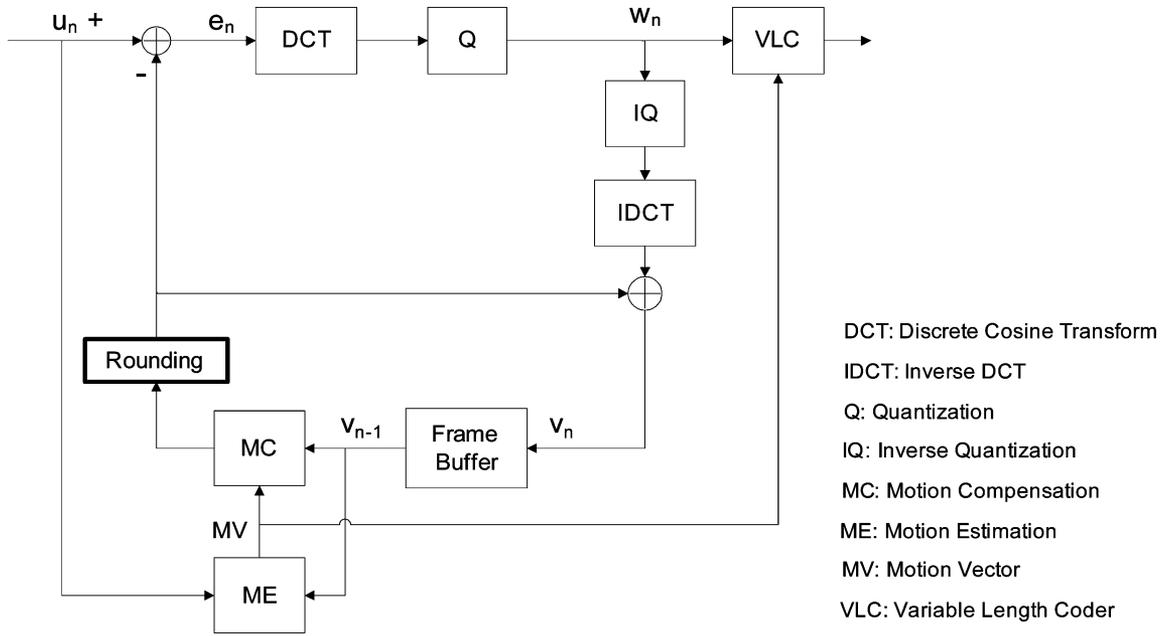


Fig. 1. Conventional encoder.

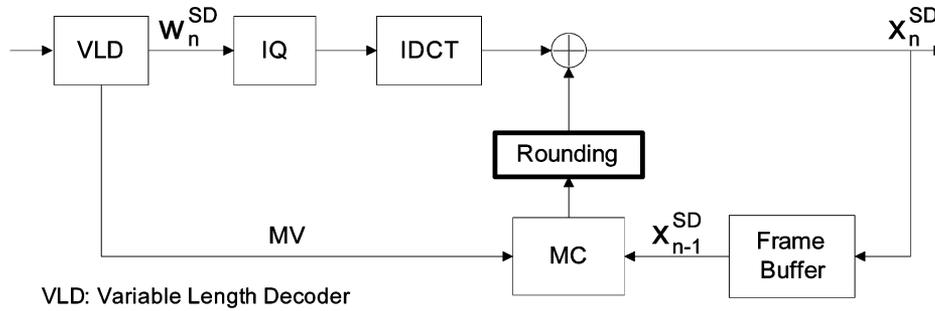


Fig. 2. Conventional decoder.

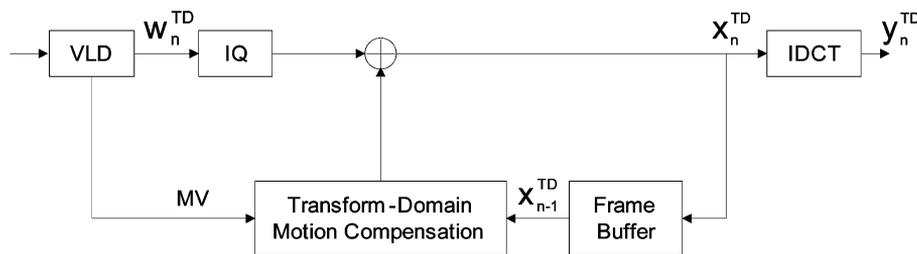


Fig. 3. Decoder with transform-domain MC.

- Intra-frame (I-frame) encoding in SE

$$w_0 = Q(DCT(u_0)) \tag{1}$$

$$v_0 = IDCT(IQ(w_0)). \tag{2}$$

- Inter-frame (P-frame) encoding in SE

$$w_n = Q(DCT(u_n - Round(MC(v_{n-1})))) \tag{3}$$

$$v_n = IDCT(IQ(w_n)) + Round(MC(v_{n-1})) \tag{4}$$

where u, v , and w denote the current frame, the reconstructed frame, and the quantized DCT coefficients

of the residual. $DCT(\cdot), IDCT(\cdot), Q(\cdot), IQ(\cdot)$, and $Round(\cdot)$ denote DCT, IDCT, quantization, inverse quantization, and rounding, respectively.

Note that the rounding operation in P-frames only occurs when a motion vector (MV) is pointing to a noninteger position; that is, MC is performed with subpel accuracy. Usually, subpel pixel values are interpolated from either two or four neighboring pixels in the reference frame. In the transform domain, this is equivalent to averaging either two or four shifted neighboring blocks of DCT coefficients in the reference frame. Since the subpel accuracy MC involves averaging, the rounding operation is unavoidable if floating point operation is not used. Later

we show that the inability to perform rounding operation in the transform domain is the major cause of the mismatch.

Fig. 2 presents the SD. The formulation for the SD is described as follows.

- Intra-frame (I-frame) decoding in SD

$$x_0^{\text{SD}} = \text{IDCT}(\text{IQ}(w_0^{\text{SD}})) \quad (5)$$

- Inter-frame (P-frame) decoding in SD

$$x_n^{\text{SD}} = \text{IDCT}(\text{IQ}(w_n^{\text{SD}})) + \text{Round}(\text{MC}(x_{n-1}^{\text{SD}})) \quad (6)$$

where the superscript “SD” represents signals of the SD.

Since the variable length coder (VLC) is lossless, the following equality is satisfied:

$$w_n = w_n^{\text{SD}}, \quad n \geq 0. \quad (7)$$

Therefore

$$x_n^{\text{SD}} = v_n, \quad \text{for } n \geq 0 \quad (8)$$

and there is no mismatch between SE and decoder.

Similar to the formulation of SD, we describe the TD (see Fig. 3) as follows:

- Intra-frame (I-frame) decoding in TD

$$y_0^{\text{TD}} = \text{IDCT}(\text{IQ}(w_0^{\text{TD}})). \quad (9)$$

- Inter-frame (P-frame) decoding in TD

$$y_n^{\text{TD}} = \text{IDCT}(\text{IQ}(w_n^{\text{TD}}) + \text{MC}^{\text{TD}}(x_{n-1}^{\text{TD}})). \quad (10)$$

In order to match TD with SE, y_n^{TD} must be equal to x_n^{SD} . However

$$\begin{aligned} y_n^{\text{TD}} &= \text{IDCT}(\text{IQ}(w_n^{\text{TD}}) + \text{MC}^{\text{TD}}(x_{n-1}^{\text{TD}})) \\ &= \text{IDCT}(\text{IQ}(w_n^{\text{TD}})) + \text{IDCT}(\text{MC}^{\text{TD}}(x_{n-1}^{\text{TD}})) \\ &\neq \text{IDCT}(\text{IQ}(w_n^{\text{SD}})) + \text{Round}(\text{MC}(x_{n-1}^{\text{SD}})). \end{aligned} \quad (11)$$

Therefore, $y_n^{\text{TD}} \neq x_n^{\text{SD}}$, and mismatch occurs. Note that if the MC and DCT/IDCT are performed in floating point operation, the rounding operation in (11) is not needed. Then the distributive property of IDCT and commutative property of MC and IDCT can be satisfied. Consequently, there is no mismatch problem. Nevertheless, for computational consideration, integer operations are usually used in conventional video encoders and decoders. Thus, the mismatch of SE-TD pair is inevitable since

$$\text{IDCT}(\text{MC}^{\text{TD}}(x_{n-1}^{\text{TD}})) \neq \text{Round}(\text{MC}(x_{n-1}^{\text{SD}})). \quad (12)$$

For an image block in the spatial 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 spatial domain is equivalent to adding 4 to the dc coefficient of the image block in the DCT (or transform)

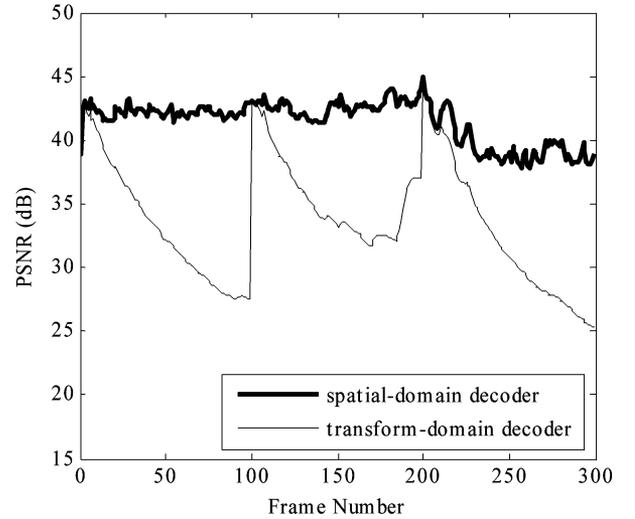


Fig. 4. PSNR curves of a SD and a TD. Note the accumulation of drift error due to rounding mismatch. (Foreman 4 Mbits/s with $N = 100$ and $M = 1$).

domain. To complete the rounding operation in the transform domain, another operation equivalent to the truncation operation in the spatial 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, and the errors would propagate and accumulate through inter-coded frames, causing severe quality degradation.

Fig. 4 shows the peak signal-to-noise ratio (PSNR) curves of two different decoders. One is a SD, and the other is a TD without an equivalent rounding operation. We can see that the PSNR drops very quickly if rounding operation is not taken care of. In Section V, we show that merely adding 4 to the dc coefficient without applying an equivalent operation of truncation causes the PSNR to drop more quickly.

III. PREVIOUS APPROACHES

In [16], the mismatch problem between spatial-domain and transform-domain codecs is analyzed, and it is pointed out that the two matching criteria can be met when floating-point operations are adopted. However, most video encoders and decoders use fixed-point or integer operations instead; thus, the truncation operation makes the two matching criteria no longer satisfied. As a result, error propagates. To reduce the effect of mismatch, a precision lifting method is proposed, which first multiplies the pixels or DCT coefficients with a precision lifting constant, and then divides that constant after the manipulations such as MC and IDCT are done. However, to ensure identical coding states at the encoder and the decoder sides, the precision lifting method should be applied at both sides, which makes this approach unattractive since we do not know whether the encoder adopts this approach. Efforts should be put on how to reproduce the same encoding states as in the encoder, no matter what domain and approach are adopted by the encoder.

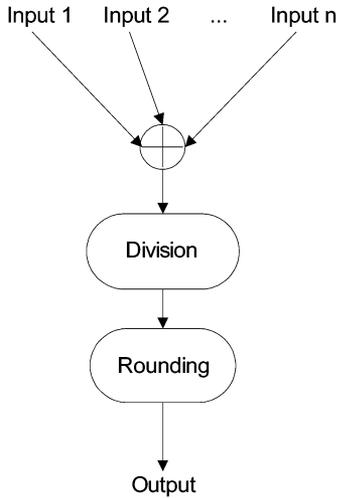


Fig. 5. Illustration of a typical averaging operation.

To average and round two pixels in the spatial domain, the sum of the pixels is added to 1 prior to dividing it by 2, which is equivalent to adding 0.5 to the average pixel value. In [11], Shanableh and Ghanbari state that since the MC-DCT employs floating-point operation, the spatial-domain rounding is performed by adding 0.5 rather than 1, which is equivalent to adding 0.25 to the average pixel value. In the DCT domain, adding a block of 0.5 corresponds to adding 4 to the dc coefficient. Similarly, when both components of the MV are of half-pel accuracy, we need to average four DCT blocks. In this case, the dc coefficient shall be adjusted by adding 8 to it prior to dividing the resulting block by 4.

Therefore, we can say that the rounding approach in [11] is to adjust the dc coefficient of the averaged block by adding 2 to it (4 divided by 2 and 8 divided by 4) regardless whether the averaging is performed on two or four blocks. We show in the following section that the minimum-error offset is not independent of the divisor. For different divisors, the optimal offsets are different.

IV. BIAS COMPENSATION

Fig. 5 shows the flow of a typical averaging operation. Suppose Input 1 to Input n and Output are all positive integers. The 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 $\{0, 0.25, 0.5, 0.75\}$, and the 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 modeled by a discrete uniform distribution. Based on the model, we describe an approach to approximating the rounding operation in the next section.

Considering the case described previously, 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

$$p_R(r) = \frac{1}{4}\delta(r + 0.25) + \frac{1}{4}\delta(r) + \frac{1}{4}\delta(r - 0.25) + \frac{1}{4}\delta(r - 0.5) \quad (13)$$

where $\delta(\cdot)$ is an impulse function. Now we want to find a value s , such that the expectation of $(R - s)^2$, 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 general block-wise data. Block \mathbf{X} is composed of the fractional values before rounding, block \mathbf{Y} is the result of rounding block \mathbf{X} , and block $\mathbf{D} = \{d_{i,j}\}_{i,j=1}^n$ is the difference between \mathbf{X} and \mathbf{Y} . The block sizes of \mathbf{X} , \mathbf{Y} , and \mathbf{D} are denoted by n . Note that, as we assumed, the elements of block \mathbf{D} are all uniformly distributed. To minimize the expected mean square error $E[(\mathbf{X} - \mathbf{Y})^2]$, we take the expectation of each element as the estimate. In the transform domain, the estimates are written as

$$\text{DCT} \begin{pmatrix} E[d_{1,1}] & \cdots & E[d_{1,n}] \\ \vdots & \ddots & \vdots \\ E[d_{n,1}] & \cdots & E[d_{n,n}] \end{pmatrix} \quad (14)$$

where $\text{DCT}(\cdot)$ denotes the DCT. Then, the transform-domain rounding operation can be approximated by adding (14) to the original block.

Therefore, if the divisor is known in advance, assuming a uniform distribution and using this approximation in the transform domain, the squared error is minimized. Moreover, the expectation of each element can be precomputed, thus making the proposed approach easy to implement.

The proposed approach is shown in Fig. 6. If the half-pel interpolation is involved in the transform-domain MC, an offset is added to the dc coefficient of the prediction block formed by the MC process. This way, the mismatch between the TD and the SE is minimized, since the encoding states of both sides are approximately equal.

V. EXPERIMENTAL RESULTS

In this section, we describe the results of experiments in which we compare the performance of the proposed approach against three existing approaches. One of these three approaches, called *without truncation*, adds a block of 0.5 to original block but does not apply the truncation operation. Another approach, *without rounding*, skips the rounding process completely. We also compare our approach with the approach in [11]. MC-DCT and IDCT in the TD are both implemented with floating point operation. Therefore, the distributive property of IDCT and the commutative property of MC and IDCT hold in the TD. Thus, the impact of the rounding operation can be seen since our goal is to approximate the rounding operation in the SE.

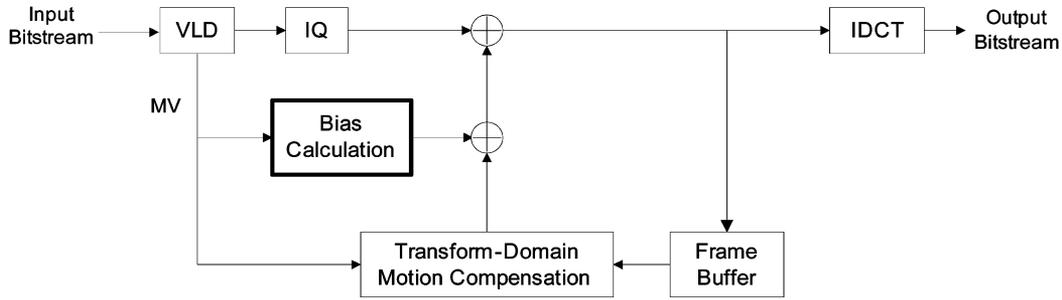


Fig. 6. Transform-domain MPEG-2 decoder with bias compensation.

TABLE I
DISTRIBUTION OF HALF-PEL INTERPOLATION WITH BOTH COMPONENTS OF MV BEING HALF-PEL RESOLUTION

Sequence	Fractional part distribution			
	0	0.25	0.5	0.75
coastguard	26.30%	24.40%	25.10%	24.30%
container	25.10%	24.60%	25.70%	24.60%
foreman	25.80%	24.50%	25.20%	24.50%
mobile	25.50%	24.70%	25.20%	24.70%
mother & daughter	25.20%	24.90%	25.10%	24.90%
paris	25.10%	24.90%	25.10%	24.90%
silent	25.30%	24.80%	25.00%	24.90%
stefan	26.10%	24.40%	25.20%	24.40%
table	26.10%	24.40%	25.20%	24.30%
weather	25.80%	24.70%	24.70%	24.80%

CIF size, 300 frames, 30fps, $N=100$, $M=1$, 4Mbits/s

TABLE II
DISTRIBUTION OF HALF-PEL INTERPOLATION WITH ONLY ONE COMPONENT OF MV BEING HALF-PEL RESOLUTION

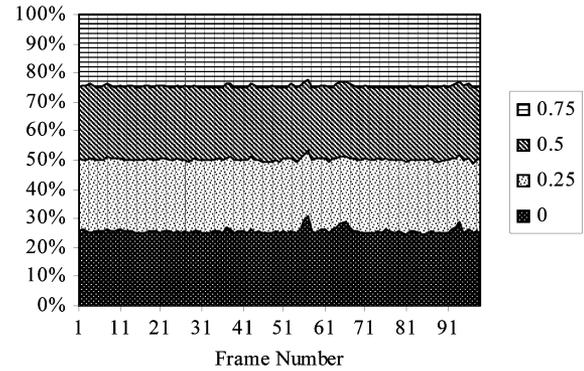
Sequence	Fractional part distribution	
	0	0.5
coastguard	53.30%	46.70%
container	50.40%	49.60%
foreman	51.10%	48.90%
mobile	50.50%	49.50%
mother & daughter	50.20%	49.80%
paris	50.20%	49.80%
silent	50.50%	49.50%
stefan	50.50%	49.50%
table	51.30%	48.70%
weather	50.50%	49.50%

CIF size, 300 frames, 30fps, $N=100$, $M=1$, 4Mbits/s

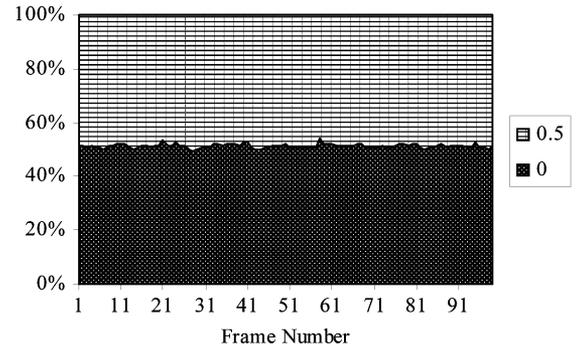
A. Verification of the Uniformly Distributed Assumption

First, we verify the assumption of uniform distribution by gathering statistics in a SD. Ten 300-frame, CIF size (352×288) sequences are encoded into 4 Mbits/s by a spatial-domain MPEG-2 TM5 encoder [22]. The GOP (Group of Picture) structure used here is (100, 1), which means that an intra frame occurs every 100 frames and no B-frame is encoded (i.e., IPPP...). During the spatial-domain decoding process, we gather the statistics of fractional value after the division at the MC part to see if the uniformly-distributed property holds.

Table I shows the distribution when both components of the MV point to half-pel position. The case of only one components



(a)



(b)

Fig. 7. Frame-by-frame distribution of the fractional value of half-pel interpolation in MC. (a) Both components of MV are in half-pel resolution. (b) Only one component of MV is in half-pel resolution. (Foreman 4 Mbits/s with $N = 100$ and $M = 1$).

of MV pointing to half-pel position is shown in Table II. We can see that the uniformly distributed assumption indeed holds. Note that even though the results in Table I and II are all averaged over 300 frames, the frame-by-frame statistic in Fig. 7 still shows an approximately uniform distribution.

Furthermore, when bilinear interpolation is used to interpolate quarter-pel pixels, we find that the fractional part generated by each divisor is uniformly distributed as well.

B. Random Number Experiment

Fig. 8 shows our second experiment, where (a) is equal to 0.5, and (b) is equal to 0.125. Three different operations are applied to a 4×4 block \mathbf{r} . Block \mathbf{y} is obtained by rounding \mathbf{r} , and serves as the benchmark representing the pixel domain result. \mathbf{y}_t and

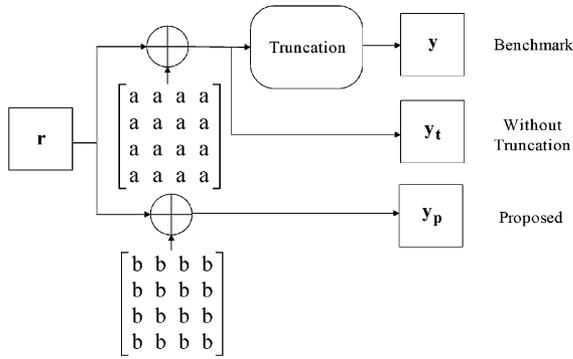


Fig. 8. Random number experiment.

TABLE III
PRECISION IN EACH APPROACH

	Approach			
	Without rounding	Without truncation	[11]	Proposed
MSE	0.1254	0.2184	0.0938	0.0781

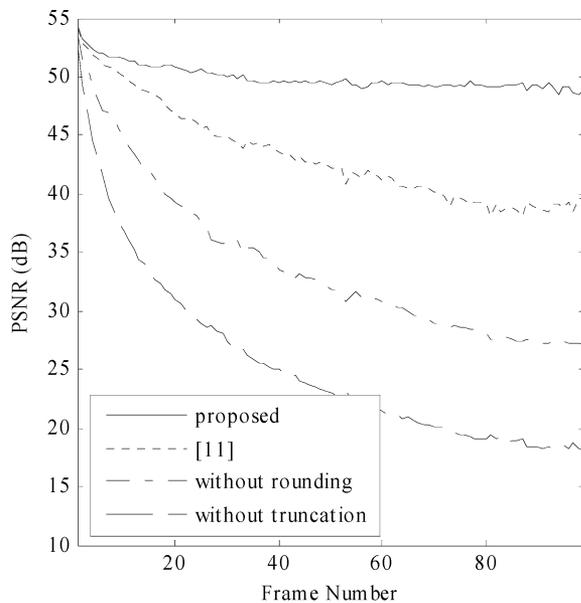


Fig. 9. PSNR curves of different approaches used to decode *Foreman* 4 Mbits/s with $N = 100$ and $M = 1$.

y_p are obtained by using *Without Truncation* approach and the proposed approach, respectively.

In the simulations, r is randomly generated, where each element in r 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 (MSE) in the spatial domain and the transform domain. We make 10000 runs and calculate the average MSE.

Table III 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 uniformly distributed assumption holds.

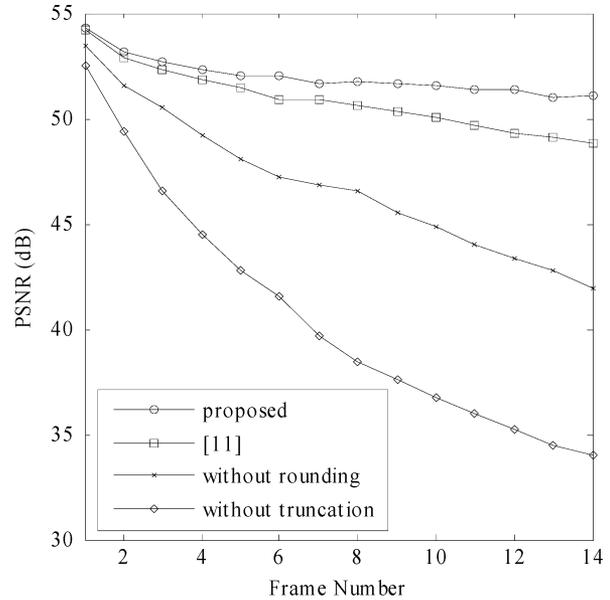


Fig. 10. PSNR curves of different approaches used to decode *Foreman* 4 Mbits/s with $N = 15$ and $M = 1$.

TABLE IV
PSNR OF THE 14TH FRAME USING DIFFERENT APPROACHES

Sequence	Approach			
	Without rounding	Without truncation	[11]	Proposed
Coastguard	39.076	34.960	48.532	48.551
Foreman	41.898	34.013	48.794	51.041
Mobile	39.892	34.472	45.740	48.020
Stefan	41.509	34.435	46.964	47.859
Tempete	41.657	34.273	49.127	50.546
Football	44.480	34.659	48.108	49.375

CIF size, 30fps, $N=15$, $M=1$, 4Mbits/s

TABLE V
PSNR OF THE 99TH FRAME USING DIFFERENT APPROACHES

Sequence	Approach			
	Without rounding	Without truncation	[11]	Proposed
Coastguard	26.843	18.524	39.036	40.552
Foreman	27.055	18.133	39.403	48.905
Mobile	25.308	17.842	36.196	40.085
Stefan	30.411	20.07	38.030	40.168
Tempete	27.487	18.028	39.890	46.416
Football	35.383	26.730	49.524	50.820

CIF size, 30fps, $N=100$, $M=1$, 4Mbits/s

C. Transform-Domain Decoder

In the third experiment, we apply the proposed method to the transform-domain MC in an MPEG-2 decoder. The transform-domain MC with half-pixel accuracy is performed by averaging either two or four motion compensated blocks. When both components of the MV have half-pixel accuracy, the divisor is 4. According to the proposed algorithm, the rounding operation can be approximated by adding 1 to the dc coefficient of the image block in the transform domain. When only one component of the MV, horizontal or vertical, has half-pixel accuracy, the divisor is 2, and the rounding operation is approximated by adding 2 to the dc coefficient of the image block in the transform domain. While in [11], the value added to the dc



Fig. 11. Subjective comparison of the 99th frame of Foreman decoded by different approaches. (a) SD. (b) Without rounding. (c) Without truncation. (d) Approach in [11]. (e) Proposed approach. (CIF size, 4 Mbits/s with $N = 100$ and $M = 1$).

coefficient is 2 whether the divisor is 2 or 4, which would not minimize the mean square error when both components of the MV have half-pixel accuracy.

Several CIF size test sequences are encoded into 4 Mbits/s bitstream by an MPEG-2 TM5 encoder. To illustrate the accumulation of drift error, a GOP structure (100, 1) is selected. If there is no rounding mismatch, the reconstructed image of the TD is identical to the one generated by the SD. Therefore, it makes sense to use the error between the decoded frames generated by the TD and the SD for PSNR calculation, as oppose to the error between the decoded frame of the TD and the original frame. Besides, since rounding mismatch only affects those blocks with subpixel MVs, only such blocks are included in the PSNR calculation. Fig. 9 is the luminance PSNR plot that shows the impact on video quality for different approaches. It can be seen that the proposed approach has the least drift error. To see the effect of GOP size, *Foreman* sequence is encoded with GOP structure (15, 1). The resulting PSNR curves are shown in Fig. 10. We can see that the error accumulates very fast even when the GOP size is small. Tables IV and V show the PSNR of the last frame in the GOP. Note that in the case of GOP size = 15, the PSNR difference between the proposed approach and [11] is only 0.02 dB for the *Coastguard* sequence. We find that this is because that there are totally only 47 macroblocks in the entire GOP that have half-pixel motion in both directions. This is very small compared to the total number of macroblocks (= 3743) with subpixel MVs. Further PSNR comparisons using the original images as reference are described in [17].

Sample frames of the decoded *Foreman* sequences are shown in Fig. 11 for subjective comparison. As we can see in Fig. 11(b),

the MC process without performing rounding operation accumulates errors, which in turn cause the picture to turn green. When the dc value is inappropriately compensated, the picture turns pink, as shown in Fig. 11(c). The coloration is the result of drift error accumulation in the chrominance components. It suggests that the drift error is more serious for the chrominance components. This is because that an integer-pixel MV for the luminance component may become a half-pixel MV for the corresponding chrominance components [23].

VI. CONCLUSION

In this paper, the rounding mismatch between spatial-domain and transform-domain video codecs is analyzed. The rounding is a basic operation for video codecs. However, because it is a nonlinear operation, the equivalent operation in the transform domain is undefined. We have presented an approximate solution to the problem. By adding an appropriate dc offset to the DCT block, our approach is able to minimize the rounding mismatch and thus reduce the error propagation through inter-coded frames. Furthermore, we have shown that the value of the dc offset depends on the divisor used for averaging the motion-compensated blocks. Our solution performs better than the previous approach that uses a constant dc bias. With this solution, the performance gap between the transform-domain and spatial-domain video codecs is significantly reduced.

It is well-known that adding a constant to an image block is equivalent to adjusting the dc value of the transform block. In the context of rounding mismatch, it is a critical issue to determine how the mismatch error can be minimized. Interestingly,

this issue has been largely ignored in the past. To our best knowledge, the solution we have developed represents the first that is able to address this issue satisfactorily with a sound theoretical basis.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their careful reviews and helpful comments, which lead to significant improvements of the paper.

REFERENCES

- [1] *Information Technology-Coding of Moving Pictures and Associated Audio for Digital Storage Media at Up to About 1.5 Mbit/s-Part 2: Video*, ISO/IEC 11172-2, 1993.
- [2] *Information Technology-Generic Coding of Moving Pictures and Associated Audio Information-Part 2: Video*, ISO/IEC 13818-2, 1995.
- [3] *Information Technology-Coding of Audio/Visual Objects*, ISO/IEC 14496-2:1999, 1999.
- [4] *Video Codec for Audio Visual Services at $p \times 64$ Kbits/s*, ITU-T Rec. H.261, 1990.
- [5] *Video Coding for Low Bit Rate Communication*, ITU-T Rec. H.263, Jan. 1998.
- [6] *Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification*, ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, 2003.
- [7] R. H. J. M. Plompen, B. F. Schuurink, and J. Biemond, "A new motion-compensated transform domain coding scheme," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1985, vol. 10, pp. 371–374.
- [8] R. H. J. M. Plompen, J. G. P. Groenveld, D. E. Boekee, and F. Booman, "The performance of a hybrid video-conferencing coder using displacement estimation in the transform domain," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1986, vol. 11, pp. 157–160.
- [9] S.-F. Chang and D. G. Messerschmitt, "Manipulation and compositing of MC-DCT compressed video," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 1, pp. 1–11, Jan. 1995.
- [10] J. Xin, C.-W. Lin, and M.-T. Sun, "Digital video transcoding," in *Proc. IEEE*, Jan. 2005, vol. 93, no. 1, pp. 84–97.
- [11] T. Shanableh and M. Ghanbari, "Hybrid DCT/pixel domain architecture for heterogeneous video transcoding," *Signal Process.: Image Commun.*, vol. 18, no. 8, pp. 601–620, Sep. 2003.
- [12] 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, no. 12, pp. 953–967, Dec. 1998.
- [13] U. V. Koc and K. J. R. Liu, "Motion compensation on DCT domain," *J. Appl. Signal Process.*, vol. 3, pp. 147–162, 2001.
- [14] N. Merhav and V. Bhaskaran, "Fast algorithms for DCT-domain image downsampling and for inverse motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 3, pp. 468–476, Jun. 1997.
- [15] J. Song and B.-L. 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.
- [16] S.-K. Oh and H. Park, "Analysis of IDCT and motion-compensation mismatches between spatial-domain and transform-domain motion compensated coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 7, pp. 835–843, Jul. 2005.
- [17] C. Chen, P.-H. Wu, and H. Chen, "A practical solution to transform-domain rounding," in *Proc. 13th Eur. Signal Process. Conf.*, Sep. 2005, pp. 487–492.
- [18] C. Chen, P.-H. Wu, and H. Chen, "MPEG-2 to H.264 transcoding," in *Picture Coding Symp.*, Dec. 2004.
- [19] P. Assuncao and M. Ghanbari, "Post-processing of MPEG2 coded video for transmission at lower bit rates," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Atlanta, GA, May 1996, vol. 4, pp. 1998–2001.
- [20] D. G. Morrison, M. E. Nilson, and M. Ghanbari, "Reduction of the bit-rate of compressed video while in its coded form," in *Proc. 6th Int. Workshop Packet Video*, 1994, pp. D17.1–D17.4.
- [21] G. Keesman, R. Hellinghuizen, F. Hoeksema, and G. Heideman, "Transcoding of MPEG bitstreams," *Signal Process.: Image Commun.*, vol. 8, no. 6, pp. 481–500, Sep. 1996.
- [22] *Test Model 5*, ISO/IEC JTC1/SC29/WG11/N0400, Apr. 1993.
- [23] J. Youn, M.-T. Sun, and J. Xin, "Video transcoder architectures for bit rate scaling of h.263 bit streams," in *Proc. 7th ACM Multimedia*, 1999, pp. 243–250.