TRANSFORM-DOMAIN WIENER FILTERING FOR H.264/AVC VIDEO ENCODING AND ITS IMPLEMENTATION

Byung Cheol Song, Nak Hoon Kim, and Kang Wook Chun

Digital Media R&D Center, Samsung Electronics Co., Ltd.
416 Maetan-3dong, Yeongtong-gu, Suwon-City, Republic of Korea
E-mail: bcsong@samsung.com

ABTRACT

For coding efficiency as well as noise reduction, efficient de-noising needs to be performed prior to video encoding. This paper proposes a transform-domain Wiener filtering scheme in an H.264/AVC video encoder. We show that the generalized Wiener filtering for each integer-transformed block is equivalent to multiplication of multiplication factor (MF) in a block with a proper filter coefficient matrix in a quantization process. Also, we implement efficiently the proposed scheme by employing several pre-determined modified MF's for quantization. Experimental results show that the proposed de-noising scheme provides outstanding coding efficiency as well as noise reduction in an H.264/AVC video encoder.

1. INTRODUCTION

Generally, input video sequences of a video encoder can be deteriorated due to various noise sources. One of main sources of contamination for video sequences is the additive noise introduced by the capture device, e.g., photonic noise in the case of charge-coupled devices (CCD) camera. Analog video signals, which are often degraded due to channel noise in the conventional analog transmission systems, need to be sometimes digitally encoded in the receiver side. Currently, many AV applications such as DVD recorder, BD recorder, and PVR can encode analog video signals, e.g., NTSC/PAL. However, noisy analog video signals are not only visually annoying, but they are also hard to be encoded efficiently owing to uncorrelated nature of noise. In order to remove as much high-frequency information as possible without compromising visual quality, effective de-noising is required.

Many de-noising schemes for noise removal in video sequences have been developed [1-5]. Among them, a spatial-domain adaptive Wiener filtering scheme [3] and a motion-compensated spatio-temporal filtering scheme [5] have good de-noising performance. The conventional de-noising schemes have been devised in the light of noise reduction itself rather than optimal combination of filtering with a video encoder. Since the noise reduction operation has been thought of as a process independent of video encoding, the de-noising schemes have been usually cascaded with video encoders. However, in the cascaded structure, a video encoder becomes computationally heavier due to the additional filtering complexity.

Kim and Ra proposed a DCT-domain noise reduction scheme (DCTNR), which is gracefully embedded with a video coder [6]. They first applied the concept of the generalized Wiener filter [7] to a video encoder. This transform-domain Wiener filtering accomplishes fast de-noising because all the processing is operated in the DCT domain simply by scaling the DCT coefficients.

Recently, the Joint Video Team (JVT) of ITU-T and MPEG experts has presented the H.264/AVC video coding standard [8], which shows significant performance gain over contemporary video coding standards such as MPEG2 and H.263. Up to now, there are no de-noising schemes embedded in an H.264 encoder. Unfortunately, it is hard to directly apply the DCTNR to the H.264 encoder. Firstly, significant computational complexity is still burdensome due to many multiplication operations during transform-domain filtering if Winograd fast integer transform is not available. Secondly, since H.264 has a new tool called intra prediction for coding efficiency of intra macroblocks (MB's), we should consider Wiener filtering for intra residual blocks.

This paper presents a transform-domain Wiener filtering scheme, which figures out the afore-mentioned problems in an H.264 encoder. We show that the generalized Wiener filtering for each integer-transformed block is equivalent to multiplication of multiplication factor (MF) in a block with a proper filtering matrix in a quantization process. Also, we provide a method to implement efficiently the proposed scheme by adaptively selecting several pre-determined MF tables according to inter/intra mode. This merging of generalized Wiener filtering into quantization

0-7803-9134-9/05/$20.00 ©2005 IEEE

process needs no computational complexity for noise reduction except only a few additional memories. Simulation results show that the proposed de-noising scheme has superior coding efficiency as well as noise reduction in an H.264 video encoder.

The organization of this paper is as follows. Section 2 presents a previous work, i.e., DCTNR. Section 3 describes the proposed algorithm. Section 4 gives simulation results. Finally, Section 5 provides conclusion.

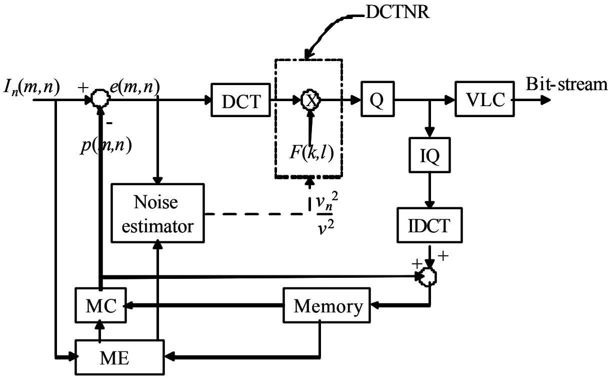

Fig. 1. The conventional video encoder employing the DCTNR [6] and the noise estimator [9].

2. PREVIOUS WORK

In Fig. 1, we can find that the whole operation for the DCTNR is equivalent to a unified scaling operation with a scaling matrix **F** in the DCT domain, which is described as follows [6]:

$$F(k,l)=\frac{1+S(k,l)\ \dfrac{v_n^2}{v^2}\ \dfrac{1}{r(k,l)}}{1+\dfrac{v_n^2}{v^2}\ \dfrac{1}{r(k,l)}} \qquad (1)$$

In Eq. (1), if (k,l) is (0,0), $S(k,l)$ is 1. Otherwise, $S(k,l)$ is 0. So, $F(k,l)$, i.e., the (k,l)-th filter coefficient is determined depending on the signal-to-noise level and $r(k,l)$. $r(k,l)$ is a normalized element on a diagonalized matrix of DCT-ed covariance, and realistic covariance estimates for mean-subtracted intra and inter blocks can be found by using some training video sequences [6]. v^2 denotes the variance of the desired noise-free data, and it is usually estimated by subtracting the noise variance (v_n^2) from the variance of an input block, i.e., v_e^2. Hence,

$$v^2=\max\{v_e^2-v_n^2,0\}. \qquad (2)$$

v_e^2 can be computed easily. So, if v_n^2 is known, $F(k,l)$ is determined. As a result, the (k,l)-th coefficient of the DCT-ed input block **E**, $E(k,l)$ is filtered simply as follows:

$$E'(k,l)=E(k,l)\times F(k,l). \qquad (3)$$

More detailed explanation about Eq. (1) is described in [6].

3. THE PROPOSED SCHEME

This paper proposes a transform-domain Wiener filtering (TDWF) embedded in an H.264 video encoder (see Fig. 2). Since this paper considers main profile H.264, the only size of integer transform is 4x4. Quantization matrix or weight matrix is not used. Assume that the H.264 video encoder framework employs a motion-compensated noise estimation algorithm for effective de-noising [9]. The noise estimator is based on a multi-resolution block-matching algorithm (HMRME) that provides high computational speed and high estimation performance concurrently [10]. We modified the HMRME suitably for the H.264 encoder.

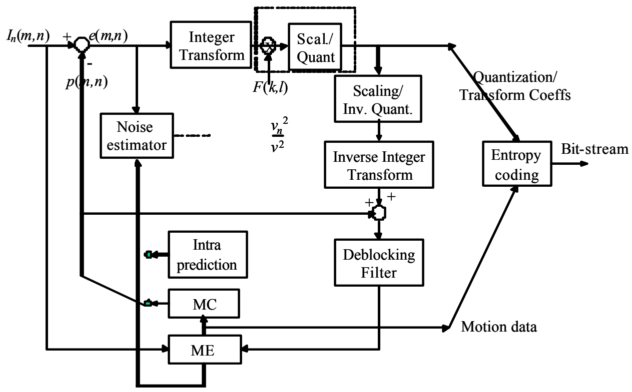

Fig. 2. Basic coding structure for H.264/AVC employing the TDWF.

As in the DCTNR, the TDWF can be accomplished by an inner-product of each integer-transformed block with a scaling matrix of Eq. (1). Note that the scaling matrix **F** has a size of 4x4. $r(k,l)$ is a normalized element on a diagonalized matrix of integer-transformed covariance. We obtain realistic covariance estimates for intra and inter blocks by using some training sequences through a method in [11]. Here, since integer transform for an intra block is performed for its residual block produced from intra prediction, $r(k,l)$ corresponding to intra MB's is derived from intra-predicted residues.

As a result, **F** of Eq. (1) is determined according to v^2/v_n^2 (SNR). While v_n^2 is updated per P-frame by the noise estimator [9], v^2 is updated on the basis of 4x4 block. So, if **F** is updated for a given SNR for each 4x4 block and its inner-product with the 4x4 transformed block is performed, the computational complexity of the TDWF shall be too burdensome. Thus, we propose to merge the inner-product of **F** and each transformed block into a quantization process.

Firstly, we need to check the quatization process of H.264 briefly. H.264 uses a scalar quantizer. There are a total of 52 values of Quantization Parameter (QP). The values of QP may be different for luma and chroma; both parameters are in the range 0-51 but QP_C is derived from QP_Y so that it QP_C is less than QP_Y for values of QP_Y above 30. A user-defined offset between QP_Y and QP_C may be signalled in a Picture Parameter Set (PPS). The basic forward quantizer operation is as follows:

$$E_q(k,l) = \text{round}\,(E(k,l) \times MF / 2^{qbits}), \quad (4)$$

where $E_q(k,l)$ is a quantized transform coefficient. *qbits* in (4) is as follows:

$$qbits = 15 + \text{floor}\,(QP/6) \quad (5)$$

MF values are derived from integer tranform (see Table I). The first 6 values of MF can be computed as follows, depending on QP and the coefficient position (k,l). For QP>5, the factors of MF remain unchanged but the divisor 2^{qbits} increases by a factor of 2 for each increment of 6 in QP. For example, $qbits$=16 for $5<QP<12$; $qbits$=17 for $11<QP<18$; and so on.

TABLE I: Multiplication factor (MF).

QP	Position: (0,0), (2,0), (0,2), (2,2)	Position: (1,1), (1,3), (3,1), (3,3)	Other positions
0	13107	5243	8066
1	11916	4660	7490
2	10082	4194	6554
3	9362	3647	5825
4	8192	3355	5243
5	7282	2893	4559

Then, Eq. (3) for a quantized residual block in an H.264 video encoder is described as follows:

$$E'(k,l) = E_q(k,l) \times F(k,l)$$
$$= \text{round}\,(E(k,l) \times MF / 2^{qbits}) \times F(k,l). \quad (6)$$

Eq. (6) is redefined as

$$E'(k,l) = \text{round}\,(E(k,l) \times MF' / 2^{qbits}), \quad (7)$$

where $MF' = MF \times F(k,l)$.

So, the TDWF can be achieved by employing modified MF, i.e., MF' in a quantization process as in Eq. (7). For efficient implementation, we confine the diversity of **F** by sub-sampling the SNR, i.e., v_n^2/v^2 values. Assume that we consider only N SNR's: SNR[0], … , SNR[N-1]. Each computed SNR is mapped into the nearest one among N SNR's. Note that we can pre-determine N modified MF tables corresponding to the sub-sampled SNR's in advance, and those modified MF tables may be stored in a memory. Therefore, for a given SNR of each block, both the TDWF and quantization of the block are performed concurrently, by loading the corresponding proper modified MF from the memory.

4. EXPERIMENTAL RESULTS

We have implemented our proposed algorithm on the JM9.2 H.264 reference software. We select the following parameters for H.264 encoding. The period of I frames is set to 15 frames without B-frames. The horizontal/vertical search ranges for motion estimation (ME) are set to [-16, +16]. Fast full search is used for ME. The number of previous frames used for inter motion search is set to 2. The initial QP for the first I frame is 32. Loop filter mode and rate control are on and R-D optimization mode is off. The first 50 frames of five well-known CIF video sequences having various motion types are used; *foreman*, *hall monitor*, *coast guard*, *container*, and *mobile*. The *mobile* sequence is encoded at 1Mbps due to high spatial complexity, and the other sequences are encoded at 600Kbps. By deliberately adding additive white Gaussian noise (AWGN) to the original test sequences, we produced noisy video sequences whose v_n^2's are set to 0, 25, and 64. The peak signal-to-noise ratio (PSNR) is employed as an evaluation measure. Coding performance of the proposed algorithm is compared with NoNR. NoNR stands for pure H.264 JM9.2 without any de-noising process.

TABLE II: Coding performance [dB].

	v_n^2	NoNR	TDWF
Foreman	64	33.41	34.66
	25	35.39	35.47
	0	36.54	36.54
Coast guard	64	29.42	29.79
	25	30.44	30.42
	0	31.00	31.00
Container	64	32.95	34.02
	25	34.86	35.21
	0	36.49	36.49
Hall monitor	64	33.72	35.17
	25	36.02	36.25
	0	36.99	36.99
Mobile	64	28.97	29.41
	25	29.98	30.07
	0	30.69	30.69

Table II compares the TDWF with NoNR by showing the PSNR values of luminance components according to various noise variances. We can find that the TDWF noticeably improves coding performance in an H.264 video encoder for noisy input sequences. The PSNR values are computed by adopting original noise-free images as references. PSNR's of the TDWF are much higher than those of the NoNR especially for v_n^2 of 64. For example, the TDWF outperforms the NoNR by about 1.5dB for *Hall monitor* sequence in case of v_n^2 =64.

In addition, Fig. 3 compares the reconstructed frame of the TDWF with that of the NoNR for *Foreman* sequence. Note that the proposed algorithm achieves almost perfect noise reduction, while the reconstructed frames by the NoNR is still very noisy.

5. CONCLUSIONS

This paper presents a transform-domain Wiener filtering (TDWF) scheme for effective H.264 video encoding. We prove that the generalized Wiener filtering for each integer-transformed block is accomplished by performing inner-product of multiplication factor with a proper scaling matrix during a quantization process. In addition, we propose to implement efficiently the TDWF by employing several pre-determined multiplication factors for quantization. Thus, we can achieve the TDWF and quantization of each block concurrently. Simulation results support the superiority of the proposed de-noising scheme in terms of coding efficiency as well as noise reduction in an H.264/AVC video encoder.

REFERENCES

[1] O. K. Al-Shaykh and R. M. Mersereau, "Lossy compression of noisy images," *IEEE Trans. Image Processing*, vol. 7, no. 12, pp. 1641-1652, Dec. 1998.

[2] N. Vasconcelos and R. Dufaux, "Pre and post-filtering for low bit-rate video coding," *Proc. Int. Conf. Image Processing*, vol. 1, pp. 291-294, Santa Barbara, California, Oct. 1997.

[3] J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice-Hall, 1990.

[4] M. Biloslavo, G. Ramponi, S. Olivieri, and L. Albani, "Joint Kalman-based noise filtering and motion-compensated video coding for low bit-rate videoconferencing," *IEEE International Conference on Image Processing*, pp. 992-995, 2000.

[5] K. J. Boo and N. K. Bose, "A motion-compensated spatio-temporal filter for image sequences with signal-dependent noise," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 287-298, June 1998.

[6] S. D. Kim and J. B. Ra, "Efficient block-based video encoder embedding a Wiener filter for noisy video sequences," *Journal of Visual Commun. Image Repres.*, vol. 14, no.1, pp. 22-40, March 2003.

[7] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice-Hall, 1989.

[8] JVT, "ISO/IEC 14496 10 Advanced Video Coding 3rd Edition," ISO/IEC JTC1/SC29/ WG11 N6540, July 2004.

[9] B. C. Song and K .W. Chun, 'Motion-compensated noise estimation for efficient pre-filtering in a video encoder," *IEEE International Conference on Image Processing*, pp. 211-214, Sept. 2003.

[10] B. C. Song and K. W. Chun, "Multi-resolution block matching algorithm and its LSI architecture for fast motion estimation in MPEG-2 video encoder," *IEEE Trans. Circ. Syst. Video Technol.*, vol. 14, no. 9, pp. 1119-1137, Sept. 2004.

[11] W. Niehsen and M. Brunig, "Covariance analysis of motion-compensated frame differences," *IEEE Trans. Circ. Syst. Video Technol.*, vol. 9, no. 4, pp. 536-539, 1999.

(a) (b) (c)

Fig. 3. Comparison of the reconstructed frames of the 1st I-frame in *Foreman* sequence: (a) Input frame (b) NoNR (PSNR = 32.01 dB) (c) TDWF (PSNR = 33.69dB).