

Optimal Block Boundary Pre/Postfiltering for Wavelet-Based Image and Video Compression

Jie Liang, *Member, IEEE*, Chengjie Tu, *Member, IEEE*, and Trac D. Tran, *Member, IEEE*

Abstract—This paper presents a pre/postfiltering framework to reduce the reconstruction errors near block boundaries in wavelet-based image and video compression. Two algorithms are developed to obtain the optimal filter, based on boundary filter bank and polyphase structure, respectively. A low-complexity structure is employed to approximate the optimal solution. Performances of the proposed method in the removal of JPEG 2000 tiling artifact and the jittering artifact of three-dimensional wavelet video coding are reported. Comparisons with other methods demonstrate the advantages of our pre/postfiltering framework.

Index Terms—JPEG 2000, pre/postfilter, three-dimensional (3-D) wavelet video coding, tiling artifact removal, wavelet transform.

I. INTRODUCTION

IT IS WELL known that DCT-based image and video coding systems exhibit annoying blocking artifacts at low bit rates. Many techniques have been proposed to resolve this problem (e.g., [1], [2], and the references therein). Among them, the lapped transform [3] generates basis functions that cover two data blocks by applying a postfilter after the DCT. This not only suppresses the blocking artifact, but also improves the compression capability.

The wavelet transform provides another solution of eliminating the blocking artifact. It is obtained by successively iterating a two-channel filter bank on its lowpass output. As a result, the wavelet transform has long low frequency basis functions and short high frequency basis functions. Blocking artifact can be removed if the wavelet transform is applied to the entire image. This, together with other properties, has made wavelets the foundation of JPEG 2000 [4].

Various three-dimensional (3-D) wavelet transform algorithms have also been proposed for video coding [5]–[10], in which another dimension of the wavelet transform is applied in the temporal direction in addition to the two-dimensional (2-D) wavelet transform within each frame. An attractive property of 3-D wavelet video coding is that it allows full scalabilities, including quality scalability, frame rate scalability, and spatial

resolution scalability. Therefore, 3-D wavelet video coding is being considered for the scalable video coding standard in MPEG-21.

However, due to the constraints of CPU and memory on different platforms, the block-based wavelet transform has to be employed in many circumstances. In image coding, this requires partitioning a large input image into some smaller pieces and encoding each separately. In JPEG 2000, each piece is called a *tile*. Although the tile size in JPEG2000 is much larger than the block size in JPEG, discontinuity known as *tiling artifact* at tile boundaries is still noticeable at low bit rates [11].

The block-based approach is more imperative in 3-D wavelet video coding due to the large amount of data involved. In this case, the video sequence is usually divided into groups of pictures (GOP), and each group is compressed by 3-D wavelet independently. This approach reduces the memory requirement, improves the robustness to transmission error, and also facilitates random access. However, it also leads to serious degradation of quality at group boundaries. This causes the periodical *jittering artifact* in video playback [12].

Two classes of methods have been proposed to reduce the tiling artifacts. The first class applies postprocessing at the decoder side. An example of this class is the method of projection onto scaling functions [11]. Although these methods can improve the visual quality, the SNR measure is usually deteriorated. In addition, postprocessing-only methods tend to introduce blurring, ringing, or other degradations.

In the second category, the problem is addressed by both the encoder and the decoder. In [13], a point-symmetric extension method is proposed to replace the conventional symmetric extension. In [14] and [15], the tile size is chosen to be odd such that the first and the last subband of each tile are always low-pass, based on the observation that high-pass coefficients at the tile boundary have larger reconstruction errors than low-pass coefficients. Since the JPEG 2000 standard requires that low-pass coefficients must be located on even canvas coordinates and high-pass coefficients on odd coordinates, it is further proposed in [16] to choose the tiles such that neighboring tiles are overlapped by one sample. This method has been included in JPEG 2000 standard as an extension to the baseline codec. However, since it is not a mandatory part of the standard, the method is not widely supported. Therefore, solutions independent of the baseline codec are highly desired. In addition, as pointed out in [11], overlapped tiles introduce a rate cost, because each tile boundary is coded twice.

To reduce the jittering artifacts in 3-D wavelet coding, a pipeline implementation is proposed in [12] to achieve the wavelet transform of the entire video sequence. However, it

Manuscript received March 7, 2004; revised November 16, 2004. This work was supported in part by the National Science Foundation (NSF) under CAREER Grant CCR-0093262, in part by NSF Grant CCF-0430869, and in part by the Simon Fraser University Startup Grant. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Charles D. "Chuck" Creusere.

J. Liang is with the School of Engineering Science, Simon Fraser University, Burnaby, BC V5A 1S6 Canada (e-mail: jiel@sfu.ca).

C. Tu is with Microsoft Corporation, Redmond, WA 98052 USA (e-mail: chentu@microsoft.com).

T. D. Tran is with the Electrical and Computer Engineering Department, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: trac@jhu.edu).
Digital Object Identifier 10.1109/TIP.2005.857244

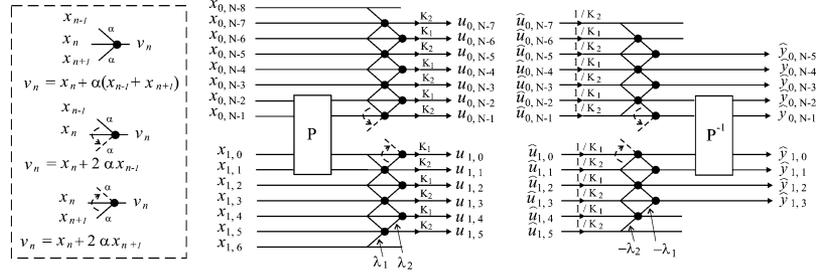


Fig. 1. Block-based wavelet transform with pre/postprocessing at block boundaries. The left figure shows the details of different lifting steps.

requires more frame buffers, and sacrifices the flexibility and robustness of the original GOP approach.

In this paper, we propose a pre/postfiltering approach, which can be viewed as the generalization of the DCT-based pre/postfiltering scheme in [17] to wavelet-based systems. In this method, a small preprocessing operator is applied at tile boundaries or GOP boundaries before wavelet compression. At the decoder side, a postprocessing operator is performed at the boundaries after wavelet decoding. The postfilter can be viewed as a deblocking filter, whereas the prefilter helps to improve the compression capability of the wavelet transform. The structure offers some attractive properties. For example, the pre/postfiltering can be implemented outside existing image and video coding systems, and the size of the pre/postfilter can be adjusted easily. In addition, although the proposed scheme requires efforts at both the encoder and decoder, we will show that postfiltering alone is able to improve the visual quality significantly at low bit rates.

The idea of applying pre/postfiltering to improve the coding performance can be traced back to [18], where a DPCM coding system with prefilter $H(z)$ and postfilter $1/H(z)$ is developed, and the optimal prefilter that maximizes the output SNR is shown to be a half-whitening filter. In [19], it is proven that the result still holds when the quantizer is replaced by a paraunitary filter bank. The result in [19] is generalized in [20] to the case where the postfilter is not the inverse of the prefilter. Prefiltering in the wavelet transform has also been studied. In [21], it is used to compute various wavelet series coefficients. The optimal design of this kind of prefilter is studied in [22] and is generalized in [23] to the multiwavelet case. However, the pre/postfilters in these papers are applied to all input samples, whereas our method only uses pre/postfilters at block boundaries.

The rest of the paper is organized as follows. In Section II, we present two algorithms to obtain the optimal filter, based on boundary filter bank and polyphase format, respectively. A fast structure is also proposed. Design examples are presented in Section III. In Sections IV and V, we apply the proposed framework to image and video coding, and demonstrate its advantages through comparison with existing methods.

II. PROBLEM FORMULATION

Fig. 1 illustrates an example of wavelet-based image/video compression with $2K \times 2K$ prefilter \mathbf{P} and postfilter \mathbf{P}^{-1} operating at tile/GOP boundaries, where $2K$ is the size of the pre/postfilter, and $x_{0,n}$ and $x_{1,n}$ ($n = 0, \dots, N-1$) are input

samples of two neighboring blocks of size N . The wavelet transform is implemented through a number of lifting and scaling steps [4], and symmetric extension is employed at block boundary. The wavelet example shown in Fig. 1 has two lifting steps. When applied to 2-D data, pre/postfiltering is first performed in all horizontal block boundaries, and then repeated at all vertical block boundaries.

Our objective is to find the optimal pre/postfilters in the rate-distortion sense. Two design approaches will be introduced in this section.

A. Boundary Filter Bank Approach

Notice that the basis functions of the DCT and the wavelet transform have different properties. In DCT, every basis function has the same length, whereas wavelet has longer low-frequency basis functions and shorter high-frequency basis functions. As a result, prefiltering prior to the DCT affects all subbands [17], but in the wavelet transform, only some boundary subbands are affected by the prefilter. This suggests that the design of wavelet prefilter should focus on the boundary region. For simplicity purpose, only one level of wavelet transform is considered here, but the result can be applied to implementations with multiple wavelet decomposition levels with satisfactory performance.

To find the optimal wavelet prefilter, we first identify the number of reconstructed pixels that are affected by the given pre/postfilters. For the example in Fig. 1 with four-point pre/postfilters and two lifting steps, it is easy to see that in block x_1 , the prefiltered boundary samples affect three wavelet coefficients after the forward transform: $u_{1,0}$, $u_{1,1}$, and $u_{1,2}$. After the inverse transform, the effect of the prefilter propagates to four reconstructed pixels: from $\hat{y}_{1,0}$ to $\hat{y}_{1,3}$. Similarly, one can verify that five reconstructed pixels in the top tile \hat{y}_0 are influenced by the prefilter.

Starting from these reconstructed pixels and tracing back, we can identify all wavelet coefficients and all input samples that contribute to these reconstructed samples. In this example, seven wavelet coefficients in the top block, and six coefficients in the bottom block are involved. They are functions of eight and seven input samples, respectively, as shown in Fig. 1.

As a result, the block boundary filter bank can be summarized as in Fig. 2, where \mathbf{F}_i and \mathbf{G}_i represent the boundary forward and inverse wavelet transforms at the two neighboring tiles, respectively. Notice that \mathbf{F}_i and \mathbf{G}_i are not square matrices. Our objective is to find the optimal pre/postfilter pair such that the

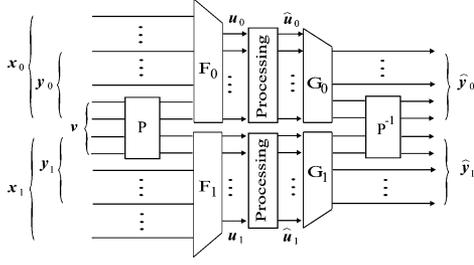


Fig. 2. Matrix representation of the block boundary filter bank in Fig. 1.

mean-squared error (MSE) of the reconstructed boundary vectors $\hat{\mathbf{y}}_0$ and $\hat{\mathbf{y}}_1$ is minimized. To this end, we first denote the boundary wavelet transform coefficients as

$$\begin{aligned} \mathbf{u} &\triangleq \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{F}_0 & 0 \\ 0 & \mathbf{F}_1 \end{bmatrix} \begin{bmatrix} \mathbf{I}_{N_0-K} & 0 & 0 \\ 0 & \mathbf{P} & 0 \\ 0 & 0 & \mathbf{I}_{N_1-K} \end{bmatrix} \begin{bmatrix} \mathbf{x}_0 \\ \mathbf{x}_1 \end{bmatrix} \triangleq \mathbf{F}\mathbf{x} \quad (1) \end{aligned}$$

where the sizes of \mathbf{x}_i , \mathbf{F}_i , and \mathbf{u}_i are $N_i \times 1$, $M_i \times N_i$, and $M_i \times 1$, respectively, and \mathbf{I}_n is the $n \times n$ identity matrix.

Let \mathbf{q}_i be the quantization noise applied to \mathbf{u}_i , the reconstruction error would be

$$\mathbf{e} = \begin{bmatrix} \mathbf{I}_{L_0-K} & 0 & 0 \\ 0 & \mathbf{P}^{-1} & 0 \\ 0 & 0 & \mathbf{I}_{L_1-K} \end{bmatrix} \begin{bmatrix} \mathbf{G}_0 & 0 \\ 0 & \mathbf{G}_1 \end{bmatrix} \begin{bmatrix} \mathbf{q}_0 \\ \mathbf{q}_1 \end{bmatrix} \triangleq \mathbf{G}\mathbf{q} \quad (2)$$

where the sizes of \mathbf{q}_i , \mathbf{G}_i , and \mathbf{y}_i are $M_i \times 1$, $L_i \times M_i$, and $L_i \times 1$, respectively.

The MSE of boundary samples can, thus, be expressed as

$$\mathcal{E} = \text{tr}\{E\{\mathbf{e}\mathbf{e}^T\}\} = \text{tr}\{\mathbf{G}\mathbf{R}_{\mathbf{q}\mathbf{q}}\mathbf{G}^T\} = \text{tr}\{\mathbf{R}_{\mathbf{q}\mathbf{q}}\mathbf{G}^T\mathbf{G}\} \quad (3)$$

where $\text{tr}\{\mathbf{A}\}$ denotes the trace of matrix \mathbf{A} .

Assuming that subband noises are uncorrelated, i.e., $\mathbf{R}_{\mathbf{q}\mathbf{q}}$ is diagonal, \mathcal{E} can be written as

$$\mathcal{E} = \sum_{k=0}^{M_0-1} \sigma_{q_0,k}^2 \|\mathbf{g}_k\|_2^2 + \sum_{k=0}^{M_1-1} \sigma_{q_1,k}^2 \|\mathbf{g}_{M_0+k}\|_2^2 \quad (4)$$

where $\sigma_{q_i,k}^2$ is the variance of the k th entry in \mathbf{q}_i , and $\|\mathbf{g}_k\|_2^2$ is the norm of the k th column of \mathbf{G} .

Define $\mathbf{R}_{\mathbf{x}\mathbf{x}}$ to be the input autocorrelation matrix and we have

$$\mathbf{R}_{\mathbf{u}\mathbf{u}} = \mathbf{F}\mathbf{R}_{\mathbf{x}\mathbf{x}}\mathbf{F}^T. \quad (5)$$

The diagonal of $\mathbf{R}_{\mathbf{u}\mathbf{u}}$ gives the variances of elements in \mathbf{u}

$$\sigma_{u_0,k}^2 = \mathbf{R}_{\mathbf{u}\mathbf{u}}(k, k) \quad \sigma_{u_1,k}^2 = \mathbf{R}_{\mathbf{u}\mathbf{u}}(M_0 + k, M_0 + k). \quad (6)$$

The variance of the quantization noise \mathbf{q} applied to \mathbf{u} can be modeled by [18]

$$\sigma_{q_i,k}^2 = C2^{-2b_{i,k}} \sigma_{u_{i,k}}^2 \quad (7)$$

where the constant C depends on the statistics of $u_{i,k}$, and $b_{i,k}$ is the number of bits allocated to the k th channel in \mathbf{u}_i .

Since the two blocks are quantized separately, the optimal bit allocation problem can be formulated as

$$\begin{aligned} \mathcal{E}_{\min} &= \text{minimize}_{\mathbf{b}_0, \mathbf{b}_1} \left(\sum_{k=0}^{M_0-1} C2^{-2b_{0,k}} \sigma_{u_{0,k}}^2 \|\mathbf{g}_k\|_2^2 \right. \\ &\quad \left. + \sum_{k=0}^{M_1-1} C2^{-2b_{1,k}} \sigma_{u_{1,k}}^2 \|\mathbf{g}_{M_0+k}\|_2^2 \right) \\ &\text{subject to } \frac{1}{M_0} \sum_{k=0}^{M_0-1} b_{0,k} = \bar{b}_0 \quad \frac{1}{M_1} \sum_{k=0}^{M_1-1} b_{1,k} = \bar{b}_1 \end{aligned} \quad (8)$$

where \bar{b}_0 and \bar{b}_1 are the average bit rates of the two blocks. This is a standard Lagrangian problem, and the optimal bit allocation can be found by writing the objective function as

$$\begin{aligned} \mathcal{L} &= \mathcal{E} + \lambda_0 \left(\frac{1}{M_0} \sum_{k=0}^{M_0-1} b_{0,k} - \bar{b}_0 \right) \\ &\quad + \lambda_1 \left(\frac{1}{M_1} \sum_{k=0}^{M_1-1} b_{1,k} - \bar{b}_1 \right). \quad (9) \end{aligned}$$

Setting $\partial\mathcal{L}/\partial b_{i,k} = 0$ yields

$$\begin{aligned} 2^{-2b_{0,k}} \sigma_{u_{0,k}}^2 \|\mathbf{g}_k\|_2^2 &= C_0 \\ 2^{-2b_{1,k}} \sigma_{u_{1,k}}^2 \|\mathbf{g}_{M_0+k}\|_2^2 &= C_1 \end{aligned} \quad (10)$$

for some constant values C_0 and C_1 . Substituting into the constraints on bit rates, C_0 and C_1 can be found to be

$$\begin{aligned} C_0 &= 2^{-2\bar{b}_0} \left(\prod_{k=0}^{M_0-1} \sigma_{u_{0,k}}^2 \|\mathbf{g}_k\|_2^2 \right)^{1/M_0} \triangleq 2^{-2\bar{b}_0} \beta_0^2 \\ C_1 &= 2^{-2\bar{b}_1} \left(\prod_{k=0}^{M_1-1} \sigma_{u_{1,k}}^2 \|\mathbf{g}_{M_0+k}\|_2^2 \right)^{1/M_1} \triangleq 2^{-2\bar{b}_1} \beta_1^2. \end{aligned} \quad (11)$$

The minimal MSE can be obtained by substituting (10) and (11) into (8).

Similar to standard subband coding problems [18], we can define the *coding gain* of the boundary filter bank with respect to the PCM scheme, where each sample is compressed directly to the bit rate of \bar{b}_i in the two blocks. The MSE of PCM is, thus

$$\mathcal{E}_{\text{PCM}} = C(L_0 2^{-2\bar{b}_0} + L_1 2^{-2\bar{b}_1}) \sigma_x^2 \quad (12)$$

where L_i is the length of \mathbf{y}_i . The coding gain of the pre/postfilter can be defined as $\gamma = \mathcal{E}_{\text{PCM}}/\mathcal{E}_{\min}$, i.e.,

$$\gamma = \frac{(L_0 2^{-2\bar{b}_0} + L_1 2^{-2\bar{b}_1}) \sigma_x^2}{M_0 2^{-2\bar{b}_0} \beta_0^2 + M_1 2^{-2\bar{b}_1} \beta_1^2} = \frac{(R_b L_0 + L_1) \sigma_x^2}{R_b M_0 \beta_0^2 + M_1 \beta_1^2} \quad (13)$$

where β_i^2 are defined in (11), and $R_b = 2^{-2(\bar{b}_0 - \bar{b}_1)}$. In this paper, $R_b = 1$ is used, i.e., we assume that the two blocks are encoded at the same bit rate.

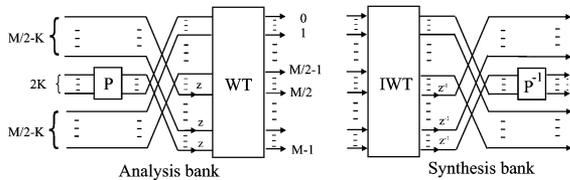


Fig. 3. Polyphase representation of the block wavelet transform with pre/postfiltering at boundaries.

TABLE I

CODING GAIN (IN DECIBLES) OF DIFFERENT BLOCK WTs WITH DIFFERENT PRE/POSTPROCESSING [FOR AR(1) INPUT WITH $\rho = 0.95$]

Block Size	16	32	64	128
	9/7 WT Only			
3-level WT	8.77	9.12	9.29	9.37
4-level WT	8.99	9.33	9.56	9.67
5-level WT	-	9.37	9.60	9.74
	9/7 WT with 4-pt Pre/Postfilter			
3-level WT	9.15	9.31	9.38	9.42
4-level WT	9.34	9.53	9.66	9.72
5-level WT	-	9.56	9.69	9.78
	9/7 WT with 8-pt Pre/Postfilter			
3-level WT	9.29	9.38	9.42	9.44
4-level WT	9.48	9.60	9.70	9.74
5-level WT	-	9.63	9.74	9.80

The optimal pre/postprocessing operator can be found by setting up an optimization program to maximize the coding gain as defined above.

B. Polyphase Approach

When all levels of wavelet decomposition of a block are viewed as a single block transform, the proposed structure in Fig. 1 can be represented by the polyphase structure in Fig. 3. This can be viewed as a generalization of the time domain lapped transform in [17] by replacing the DCT with the wavelet transform. The coding gain of this structure can be obtained by the following classical definition [18], [19]:

$$\gamma = \sigma_x^2 / \left(\prod_{k=0}^{M-1} \sigma_{u_k}^2 \|g_k\|_2^2 \right)^{1/M} \quad (14)$$

where $\sigma_{u_k}^2$ is the variance of the k th subband signal, and $\|g_k\|_2^2$ is the norm of the k th synthesis subband filter.

Notice that all subbands of a block are involved in (14), whereas only subbands near the block boundary are used in (13). Thus, optimization with (13) is much faster than the polyphase one, especially for large block size. Optimization results show that the solutions of the two methods are very similar, although only one level of WT is considered in the boundary filter bank approach. However, the polyphase method can reveal some important relationships among the size of the pre/postfilter, the block size and the levels of WT, as will be discussed in Section III-A.

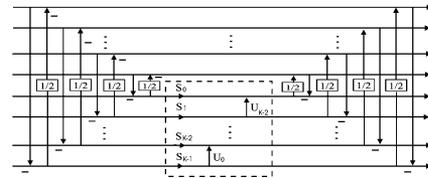


Fig. 4. Fast prefilter structure.

TABLE II

OPTIMAL PARAMETERS AND RATIONAL APPROXIMATIONS FOR THE FAST PRE/POSTFILTERS IN FIG. 4

P	S_0	S_1	S_2	S_3	U_0	U_1	U_2
2×2	1.76	-	-	-	-	-	-
	7/4	-	-	-	-	-	-
	2	-	-	-	-	-	-
4×4	1.63	1.30	-	-	0.55	-	-
	3/2	5/4	-	-	1/2	-	-
	2	1	-	-	1/2	-	-
6×6	1.65	1.33	1.17	-	0.17	0.63	-
	3/2	5/4	9/8	-	1/8	1/2	-
	2	1	1	-	1/4	1/2	-
8×8	1.66	1.35	1.19	1.13	0.10	0.21	0.63
	5/3	4/3	6/5	9/8	1/8	1/4	5/8
	2	1	1	1	1/8	1/4	1/2

C. Efficient Structure for Pre/Postfilters

In addition to coding gain, other properties such as fast computation and modular structure are often desirable for a filter bank in practical image and video coders. In the pre/postprocessing of the wavelet transform, these requirements impose further constraints on the pre/postfilters. In this paper, we propose the following structure:

$$\mathbf{P} = \frac{1}{2} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix} \quad (15)$$

where \mathbf{J} is the $K \times K$ reversal identity matrix. This structure is identical to that in the time domain lapped transform [17]. All degrees of freedom in this structure lie in the $K \times K$ matrix \mathbf{V} , which can be optimized for the coding gain.

The structure in (15) enjoys the following properties. First, the prefilter has no effect on an input vector \mathbf{c} whose entries are all identical, i.e., $\mathbf{P}\mathbf{c} = \mathbf{c}$. Therefore, no nonzero high-frequency wavelet coefficient is created for constant inputs. Second, the prefilter satisfies $\mathbf{P} = \mathbf{J}_{2K}\mathbf{P}\mathbf{J}_{2K}$; thus, it provides the same boundary processing to both blocks.

D. Postprocessing-Only Approach

Reducing blocking artifacts with only postprocessing is desired in some scenarios, especially when the user does not have access to the encoder. In Section IV, we shall show that the postfilter designed above can be directly applied to improve the visual quality, especially at low bit rates. This is because the effect of the prefiltering is essentially eliminated by the large quantization step in this case.

TABLE III
JPEG 2000 IMAGE CODING EXAMPLES WITH A TILE SIZE OF 64×64 AND DIFFERENT WTS AND PRE/POSTFILTERS

bits/pixel (bpp)	0.1	0.25	0.5	1.0	1.5	4.0	0.1	0.25	0.5	1.0	1.5	4.0
	Barbara						Boat					
9/7 WT only	21.70	25.97	30.11	35.08	38.14	43.62	23.82	28.70	32.74	37.56	40.43	46.03
4 × 4 P, 8-bit data	21.86	26.06	30.22	35.22	38.22	43.57	23.98	28.82	32.88	37.68	40.54	46.33
8 × 8 P, 8-bit data	21.89	25.96	30.19	35.12	38.07	43.23	23.91	28.77	32.85	37.65	40.53	46.33
	Goldhill						Lena					
9/7 WT only	24.88	28.81	31.79	35.30	37.84	46.11	25.07	31.30	35.41	38.79	40.80	46.25
4 × 4 P, 8-bit data	24.97	28.90	31.86	35.40	37.93	46.42	25.37	31.57	35.64	39.02	41.14	47.55
8 × 8 P, 8-bit data	24.99	28.93	31.88	35.47	38.10	48.02	25.47	31.52	35.47	38.72	40.68	45.92
	Barbara						Boat					
5/3 WT only	21.28	25.09	28.88	33.69	36.74	42.32	23.32	28.18	32.06	36.40	38.81	48.34
4 × 4 P, 9-bit data	21.33	25.07	28.90	33.75	36.81	42.43	23.34	28.18	32.09	36.51	39.02	46.34
8 × 8 P, 9-bit data	21.33	25.09	28.89	33.71	36.71	42.66	23.44	28.22	32.13	36.60	39.09	47.75
	Goldhill						Lena					
5/3 WT only	24.47	28.37	31.30	34.77	37.27	46.40	24.41	30.46	34.45	37.66	39.36	46.41
4 × 4 P, 9-bit data	24.51	28.42	31.36	34.84	37.42	46.59	24.59	30.58	34.46	37.56	39.19	46.40
8 × 8 P, 9-bit data	24.50	28.41	31.31	34.72	37.15	46.04	24.66	30.59	34.46	37.41	38.92	47.09

III. DESIGN EXAMPLES

A. Coding Gains of Pre/Postfiltered Block WT

Table I tabulates the optimized coding gains of the filtered block wavelet transform with different block sizes, iteration levels and pre/postfilter sizes. The coding gains are obtained by the polyphase method in Fig. 3. The 9/7 wavelet in JPEG 2000 is used. The input is assumed to follow an AR(1) model with correlation coefficient $\rho = 0.95$. The matrix \mathbf{V} in (15) is modeled by a full biorthogonal matrix.

It can be seen from Table I that the coding gain of a filtered block transform is equivalent to that of a unfiltered system with much larger block. For example, with eight-point pre/postfiltering, the coding gains of three-level 9/7 wavelet with a block size of 16 and 32 become comparable to that of the wavelet transform with a block size of 64 and 128, respectively. Therefore, this method is very useful to improve the performance of resource-constrained systems.

B. Design Examples for Fast Pre/Postfilter

Experimental results show that when $K < 5$, the $K \times K$ optimal matrix \mathbf{V} in (15) can be approximated very well by the fast structures in Fig. 4, which involves K scaling factors and $K - 1$ lifting steps. The structure is similar to that in the time domain lapped transform [17]. Some optimized parameters for Fig. 4 are listed in Table II. Various rational approximations that allow fixed-point implementation are also presented. Notice that those with integer scaling parameters enable lossless compression, and the resulting image/video coding system still enjoys the lossy-to-lossless compression property if invertible wavelet transforms such as the 5/3 wavelet are used.

IV. APPLICATIONS IN IMAGE COMPRESSION

In this section, we demonstrate the performance of pre/post-filtering in the removal of tiling artifact in JPEG 2000, and compare the results with that of the overlapped tiling method in

[16]. Kakadu version 3.4 [24] is used to perform the JPEG 2000 operation.

Since the prefilter is not orthogonal, the dynamic range of the input is increased slightly. For 8-bit image samples, the filtered boundary signals usually need 9 bits to represent. Two options can be selected to integrate the prefiltering and the JPEG 2000 engine. One way is to clip the filtered result to be with in the range of $[0, \dots, 255]$. The other is to subtract 128 from the prefiltered image and encode it as a 9-bit image. This method is necessary for lossless coding. The first method introduces some clipping errors, whereas coding performance is sacrificed slightly by the second method. Nevertheless, the following results show that these drawbacks are usually more than compensated by the gain of the proposed framework.

Table III lists the PSNR results of four test images for bit rates ranging from 0.1 bits/pixel (bpp) to 4.0 bpp. Five levels of 9/7 or 5/3 wavelet transform are used in all cases. The tile size is chosen to be 64×64 so that we can compare with the overlapped tiling method in [16]. The lossless four-point and eight-point filtering configurations in Table II are used for 5/3 wavelet and lossy integer configurations in Table II are used for 9/7 wavelet.

Table III shows that in most cases, the proposed method generates higher PSNR than the baseline JPEG 2000 no matter whether the filtered data are treated as 8 bit or 9 bit. Although the PSNR gain is generally less than 0.4 dB, there are some special cases where the improvement can be more than 1 dB, as highlighted in Table III (1.91 dB for Goldhill at 4.0 bpp and 1.3 dB for Lena at 4.0 bpp).

As a comparison, some PSNR results with the Lena test image are reported in [16] for bit rates from 0.1–1.5 bpp. The overlapped tiling method in it produces a gain of 0.3–0.5 dB when the bit rate is less than 0.25 bpp, but no gain is reported for rates higher than 0.5 bpp.

We next focus on the performance of the proposed method at tile boundaries. Fig. 5 shows the decoded image Lena at



Fig. 5. Portions of decoded image Lena at 0.206 bpp. (a) Kakadu only: 30.06 dB. (b) Kakadu with eight-point pre/postfiltering: 30.32 dB. (c) Kakadu with eight-point postfiltering only: 30.12 dB.

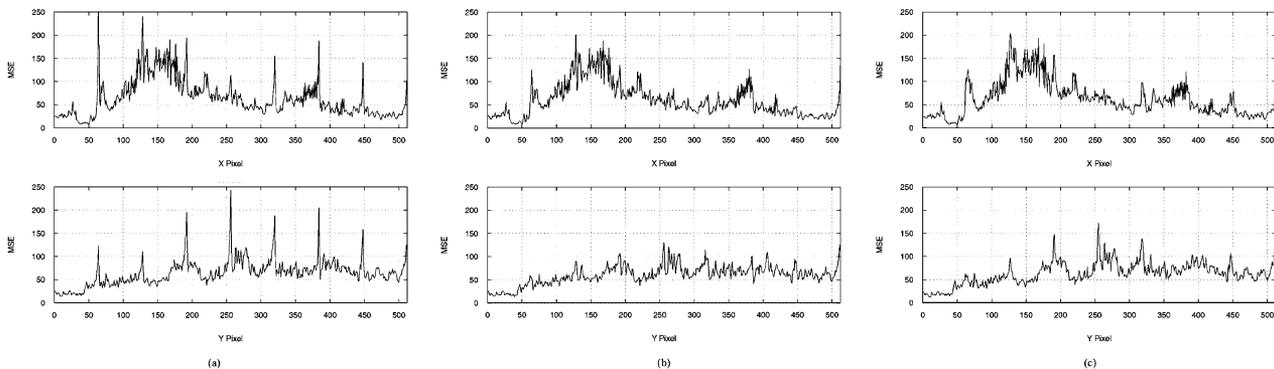


Fig. 6. Average reconstruction error for each row and each column of image Lena at 0.206 bpp and a tile size of 64×64 . (a) Kakadu only. (b) Kakadu with eight-point pre/postfiltering. (c) Kakadu with eight-point postfiltering only.

0.206 bpp with different methods. Fig. 5(a) is the Kakadu result with a PSNR of 30.06 dB. To ensure fair comparison, this example is designed to have the same PSNR as the 0.25 bpp result in [16], where JPEG 2000 verification model (VM) 8.0 is used. Tiling artifact can be observed clearly in Fig. 5(a). In Fig. 6(a), we plot the average reconstruction error over all rows and all columns of this image, which clearly indicates that the error is much higher at tile boundaries.

Fig. 5(b) is obtained with eight-point pre/postfiltering. The overall PSNR is improved to 30.32 dB. The corresponding average error in Fig. 6(b) verifies that the reconstruction error near tile boundaries has been reduced significantly, which translates to much more pleasant visual quality.

The results in Figs. 5(b) and 6(b) have striking similarities to those in [16], where a PSNR gain of 0.3 dB is reported. This suggests that the proposed method is as efficient as the overlapped method in [16] for low bit rates. Our method can also yield better coding performance at high bit rates. Besides, it does not require any change to the baseline JPEG 2000 implementation. This makes it an attractive choice for applications where the user could not change the existing JPEG 2000 software or hardware implementations.

Although the proposed framework requires both prefilter and postfilter, postprocessing alone is actually effective enough to reduce the tiling artifact at low bit rates, as demonstrated in Figs. 5(c) and 6(c), where prefiltering is bypassed and only post-

filtering is applied. However, the PSNR is only 0.06 dB better than the JPEG 2000 result.

V. APPLICATION IN 3-D WAVELET VIDEO CODING

In this section, we apply the fast pre/postprocessing solutions to reduce the jittering artifact in 3-D wavelet video coding, and compare with the global temporal WT method in [12]. Various pre/postfiltering configurations are tested, and coding results of some test sequences are reported in Table IV. The GOP size is chosen to be 16, and three levels of 9/7 wavelet transform are performed in each dimension. The lossless 9/7 wavelet as given in [25] is used. 3-D SPIHT is then used to encode the wavelet coefficients [8].

For comparison purposes, we implement the global WT method as in [12], which first obtains the global temporal WT by a pipeline algorithm. The global WT is then divided into GOPs in the wavelet domain and each GOP is coded with 3-D SPIHT to obtain a fair comparison with the proposed method (3-D ESCOT is used in [12]).

Table IV shows that compared with the original GOP approach, the global WT method improves the PSNR by 0.38 dB, on average. This is consistent with the 3-D ESCOT results in [12]. The proposed GOP WT with six-point filter yields a competitive gain of 0.25 dB, given that it is still a GOP-based solution. Notice that the global WT method is not always better than the filtered GOP approach.

TABLE IV
VIDEO CODING RESULTS (PSNR IN DECIBELS) WITH 9/7 WAVELET-BASED 3-D SPIHT AND VARIOUS PRE/POSTFILTERING

Comp. Ratio	GOP SPIHT	Global SPIHT	SPIHT & lossless 2-pt Filter	SPIHT & lossless 8-pt Filter	SPIHT & lossy 4-pt Filter	SPIHT & lossy 6-pt Filter
Akiyo (QCIF, 288 frames)						
30 : 1	41.23	41.54	41.35	41.36	41.47	41.55
60 : 1	38.25	38.72	38.39	38.45	38.48	38.53
120 : 1	33.62	33.94	33.70	33.74	33.75	33.78
Claire, QCIF, 480 frames)						
30 : 1	41.93	42.25	42.09	42.07	42.24	42.31
60 : 1	39.37	39.88	39.56	39.63	39.66	39.72
120 : 1	35.33	35.74	35.47	35.56	35.56	35.60
Suzie (QCIF, 144 frames)						
30 : 1	37.60	37.89	37.71	37.72	37.78	37.79
60 : 1	35.26	35.69	35.34	35.39	35.43	35.44
120 : 1	32.80	33.25	32.89	32.92	32.95	32.96
Foreman (QCIF, 400 frames)						
30 : 1	32.06	32.40	32.18	32.18	32.25	32.27
60 : 1	29.31	29.65	29.41	29.45	29.48	29.51
120 : 1	26.81	27.19	26.91	26.96	26.97	26.99
Average	35.30	35.68	35.42	35.45	35.50	35.55

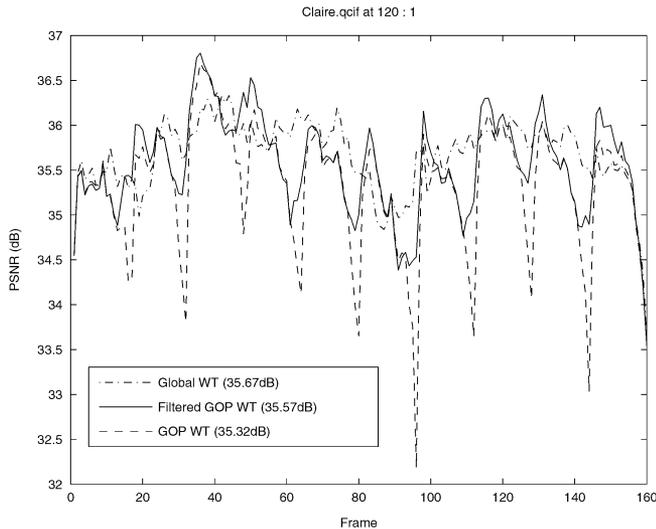


Fig. 7. Performance comparison of 3-D SPIHT video coding with global temporal WT, GOP WT, and GOP WT with six-point pre/postfilter.

The three methods are compared in Fig. 7 for the first 160 frames of the QCIF Claire test sequence coded at a compression ratio of 120:1. The original GOP method experiences serious PSNR drop at GOP boundaries, which causes the annoying periodical jittering artifact. After pre/postfiltering, the PSNR can be improved by 1.5–2.5 dB at GOP boundaries, and the jittering artifact is, therefore, reduced significantly.

Although the global method achieves higher PSNR, it introduces some other problems. First of all, more buffers are required, which implies longer latency. For example, it needs 29 and 61 frame buffers for three-level and four-level 9/7 wavelet, respectively. The required buffer in our method is simply $M+K$ for a GOP size of M and prefilter size of $2K$. Therefore, only 19

frames are needed when $M = 16$ and $K = 3$. Notice that our buffer requirement is independent of the level of wavelet iterations. Second, since the global method divides the global WT into different GOPs in the wavelet domain, any error in one GOP will propagate to neighboring GOPs after the inverse wavelet transform. This poses more challenges to operations such as random access and error concealment, especially for low-end platforms.

VI. CONCLUSION

In this paper, we present a block boundary pre/postprocessing framework for wavelet-based image and video coding. Two algorithms are proposed to obtain the optimal filters, and near-optimal fast implementation is developed. The applications in JPEG 2000 show that at low bit rates, the proposed method is comparable to the overlapped tiling method in the removal of tiling artifact. Moreover, the postfilter alone is sufficient to improve the visual quality. At high bit rates, our method can also improve the coding performance whereas the overlapped tiling method does not. Since our method does not need any change to the baseline JPEG 2000, it is very useful when the user does not have access to the existing software or hardware implementation.

When applied in 3-D wavelet video coding, the proposed method significantly reduces the jittering artifact. It also preserves the simplicity, flexibility and robustness of the original GOP-based approach. Therefore, it is very promising for resource-constrained platforms and real-time applications.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their suggestions which significantly enhanced the quality of the

revised paper. They would also like to thank Dr. M. Adams of the University of Victoria and Dr. K. Berkner of Ricoh Innovations, Inc., for their valuable comments.

REFERENCES

- [1] H. Paek, R.-C. Kim, and S.-U. Lee, "On the POCS-based postprocessing technique to reduce the blocking artifacts in transform coded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 3, pp. 358–367, Jun. 1998.
- [2] T. Chen, H. R. Wu, and B. Qiu, "Adaptive postfiltering of transform coefficients for the reduction of blocking artifacts," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 5, pp. 594–602, May 2001.
- [3] H. S. Malvar, *Signal Processing with Lapped Transforms*. Norwood, MA: Artech House, 1992.
- [4] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards, and Practices*. Norwell, MA: Kluwer, 2002.
- [5] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 572–589, May 1994.
- [6] J.-R. Ohm, "Three-dimensional subband coding with motion compensation," *IEEE Trans. Image Process.*, vol. 3, no. 5, pp. 559–571, May 1994.
- [7] S.-J. Choi and J. W. Woods, "Motion-compensated 3-D subband coding of video," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 155–167, Feb. 1999.
- [8] B.-J. Kim, Z. Xiong, and W. A. Pearman, "Low bit-rate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 12, pp. 1365–1374, Dec. 2000.
- [9] J. Xu, Z. Xiong, S. Li, and Y. Zhang, "3-D embedded subband coding with optimal truncation (3-D ESCOT)," *J. Appl. Comput. Harmon. Anal.*, vol. 10, pp. 290–315, May 2001.
- [10] S.-T. Hsiang and J. W. Woods, "Embedded video coding using invertible motion compensated 3-D subband/wavelet filter bank," *Signal Process., Image Commun.*, vol. 16, pp. 705–724, May 2001.
- [11] K. Berkner and E. Schwartz, "Removal of tile artifacts using projection onto scaling functions for JPEG 2000," in *Proc. IEEE Conf. Image Processing*, vol. 1, Rochester, NY, Sep. 2002, pp. 373–376.
- [12] J. Xu, Z. Xiong, S. Li, and Y.-Q. Zhang, "Memory-constrained 3-D wavelet transform for video coding without boundary effects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 9, pp. 812–818, Sep. 2002.
- [13] I. Kharitonenko, X. Zhang, and S. Twelves, "A wavelet transform with point-symmetric extension at tile boundaries," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1357–1364, Dec. 2002.
- [14] J. Wei, M. Pickering, M. Frater, and J. Arnold, "A new method for reducing boundary artifacts in blocking-based wavelet image compression," *Proc. SPIE*, vol. 4067, pp. 1290–1295, 2000.
- [15] J. Wei, M. Pickering, M. Frater, J. Boman, and J. Arnold, "A new method for boundary artefact reduction in JPEG 2000," in *Proc. IEEE Conf. Image Processing*, vol. 3, 2001, pp. 772–775.
- [16] J. Wei, M. Pickering, M. Frater, J. Boman, J. Arnold, and W. Zeng, "Boundary artifact reduction using odd tile length and the low pass first convention (OTLPF)," presented at Proc. SPIE Conf. Applications of Digital Image Process. [Online]. Available: <http://citeseer.ist.psu.edu/464234.html>
- [17] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-processing," *IEEE Trans. Signal Process.*, vol. 51, no. 6, pp. 1557–1571, Jun. 2003.
- [18] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [19] I. Djokovic and P. P. Vaidyanathan, "On optimal analysis/synthesis filters for coding gain maximization," *IEEE Trans. Signal Process.*, vol. 44, no. 5, pp. 1276–1279, May 1996.
- [20] J. Tuqan and P. P. Vaidyanathan, "Statistically optimum pre- and post-filtering in quantization," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 6, pp. 1015–1031, Dec. 1997.
- [21] M. J. Shensa, "The discrete wavelet transform: Wedding the a trous and Mallat algorithms," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.
- [22] X.-G. Xia, C.-C. J. Kuo, and Z. Zhang, "Wavelet coefficient computation with optimal prefiltering," *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 2191–2197, Aug. 1994.
- [23] X.-G. Xia, J. S. Geronimo, D. P. Hardin, and B. W. Suter, "Design of prefilters for discrete multiwavelet transforms," *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 25–35, Jan. 1996.
- [24] Kakadu Version 3.4, D. Taubman. (2001). [Online]. Available: <http://www.kakadusoftware.com>
- [25] Z. Xiong, X. Wu, S. Cheng, and J. Hua, "Lossy-to-lossless compression of medical volumetric data using three-dimensional integer wavelet transforms," *IEEE Trans. Med. Imag.*, vol. 22, no. 3, pp. 459–470, Mar. 2003.



Jie Liang (S'99–M'04) received the B.E. and M.E. degrees from Xi'an Jiaotong University, China, in 1992 and 1995, respectively, the M.E. degree from the National University of Singapore in 1998, and the Ph.D. degree from The Johns Hopkins University, Baltimore, MD, 2003.

He is now an Assistant Professor at the School of Engineering Science, Simon Fraser University, Burnaby, BC, Canada. From June 2003 to May 2004, he was with the Video Codec Group, Microsoft Digital Media Division, Redmond, WA. His research

interests include multirate signal processing, image/video compressions, and digital communications.



Chengjie Tu (S'02–M'04) received the B.E. and M.E. degrees from University of Science and Technology of China in 1994 and 1997, respectively, and the M.S.E. and Ph.D. degrees from The Johns Hopkins University, Baltimore, MD, in 2002 and 2003, respectively.

He is currently with the Video Codec Group, Microsoft Digital Media Division, Redmond, WA. His current research interests include image/video compression, multirate signal processing, and error control and concealment for image/video

communication.



Trac D. Tran (S'94–M'98) received the B.S. and M.S. degrees from the Massachusetts Institute of Technology, Cambridge, in 1993 and 1994, respectively, and the Ph.D. degree from the University of Wisconsin, Madison, in 1998, all in electrical engineering.

In July 1998, he joined the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, where he is currently an Associate Professor. His research interests are

in the field of digital signal processing, particularly in multirate systems, filter banks, transforms, wavelets, and their applications in signal analysis, compression, processing, and communications. He was the Co-Director (with Prof. J. L. Prince) of the 33rd Annual Conference on Information Sciences and Systems, Baltimore, MD, in March 1999. In the summer of 2002, he was an ASEE/ONR Summer Faculty Research Fellow at the Naval Air Warfare Center Weapons Division (NAWCWD), China Lake, CA.

Dr. Tran received the National Science Foundation CAREER award in 2001. He currently serves as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is also a member of the IEEE Technical Committee on Signal Processing Theory and Methods (SPTM TC).