Undersampled Boundary Pre-/Postfilters for Low Bit-Rate DCT-Based Block Coders

Lu Gan, *Member, IEEE*, Chengjie Tu, *Member, IEEE*, Jie Liang, *Member, IEEE*, Trac D. Tran, *Member, IEEE*, and Kai-Kuang Ma, *Senior Member, IEEE*

***Abstract*—It has been well established that critically sampled boundary pre-/postfiltering operators can improve the coding efficiency and mitigate blocking artifacts in traditional discrete cosine transform-based block coders at low bit rates. In these systems, both the prefilter and the postfilter are square matrices. This paper proposes to use *undersampled* boundary pre- and postfiltering modules, where the pre-/postfilters are rectangular matrices. Specifically, the prefilter is a "fat" matrix, while the postfilter is a "tall" one. In this way, the size of the prefiltered image is smaller than that of the original input image, which leads to improved compression performance and reduced computational complexities at low bit rates. The design and VLSI-friendly implementation of the undersampled pre-/postfilters are derived. Their relations to lapped transforms and filter banks are also presented. Two design examples are also included to demonstrate the validity of the theory. Furthermore, image coding results indicate that the proposed undersampled pre-/postfiltering systems yield excellent and stable performance in low bit-rate image coding.**

***Index Terms*—Discrete cosine transform (DCT), downsampling, interpolation, low bit-rate coding, pre-/postfiltering, pre-/postprocessing, undersampled.**

I. Introduction

BLOCK coders based on the *discrete cosine transform* (DCT) [1] are widely used in image and video compression. Their success stems from the DCT's excellent energy compaction capability, low computational complexity, and a high degree of flexibility. However, the main disadvantage of block coders is that the correlation between neighboring blocks has not been taken into account, which leads to suboptimal coding efficiency and the manifestation of blocking artifacts at low bit rates. To overcome these shortcomings, several postprocessing algorithms have been proposed to mitigate the blocking artifacts [2]–[11]. These algorithms work well for medium to high bit-rate compression, but most of them fail to produce satisfactory results at low bit rates. Recently, several works have been proposed to integrate both pre- and postprocessing to improve low bit-rate coding performance. Among them, two simple and effective schemes include 1) using boundary pre-/postfilters [7], [8] and 2) using the downsampling/interpolation techniques [12], [13]. Since the pre-/post-operators are completely outside the DCT and the *inverse* DCT (IDCT), both approaches can easily achieve the compliance to the DCT-based standards (e.g., the JPEG) with modest modifications. In what follows, we will briefly review these two schemes. For simplicity, we will only consider the pre-/postprocessing of 1-D signals. The 2-D signals can be handled by processing rows and columns separately.

As shown in Fig. 1(a), the first approach [7], [8] adds a boundary prefilter $\mathbf{P}$ before the DCT and a boundary postfilter $\mathbf{T}$ after the IDCT, respectively. $\mathbf{P}$ and $\mathbf{T}$ are the exact inverse of each other and they are of size $N \times N$ ($N = 2n$) each, which is the same as that of the DCT and the IDCT. The prefilter $\mathbf{P}$ is utilized to decorrelate the inter-block correlations, while the postfilter $\mathbf{T}$ is applied to reduce the quantization mismatches along block boundaries. It was demonstrated in [14] and [15] that by turning on boundary pre-/postfiltering and optimizing adaptive context-based entropy coding appropriately, a DCT-based block coder can achieve competitive coding performance as the wavelet-based JPEG2000 coder [16], [17] at a much lower computation cost. In fact, boundary prefilter combined with the DCT performs the time-domain lapped transform (LT) [7], [8], which have been well known to be effective in image processing [18]–[20]. The system in Fig. 1(a) can be also viewed as an N-channel critically sampled linear-phase filter bank (FB) with each filter of the same length $L = 2N$.

The second approach through the downsampling and interpolation techniques is shown in Fig. 1(b). Here, the input signal is first filtered through a decimation filter $H_D(z)$ and downsampled by a rational factor k ($k > 1$). The low-resolution signal is then passed through the block codec. At the output of the IDCT, the signal is upsampled and convoluted by the interpolation filter $H_I(z)$ to yield the reconstructed image. As the size of the input image to the block coder is only $(1/k)^2$ of the original image, this approach leads to significant complexity reduction of the coding/decoding processes. As shown in [12], by choosing an appropriate decimation factor k, both the PSNR and the visual quality of reconstructed images can be substantially improved. In addition, even better performance can be achieved by using optimal decimation filter $H_D(z)$ and interpolation filter $H_I(z)$ in the least square sense [13].

To take advantages of both approaches, an intuitive way is the cascading approach as shown in Fig. 1(c), where the down-

Manuscript received May 5, 2005; revised July 31, 2006. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dimitri Van De Ville.

L. Gan is with the Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool, L69 3GJ, U.K. (e-mail: ganlu@ieee.org).

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

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

T. D. Tran is with the Electrical and Computer Engineering department, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: trac@jhu.edu).

K.-K. Ma is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: ekkma@ntu.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2006.888344

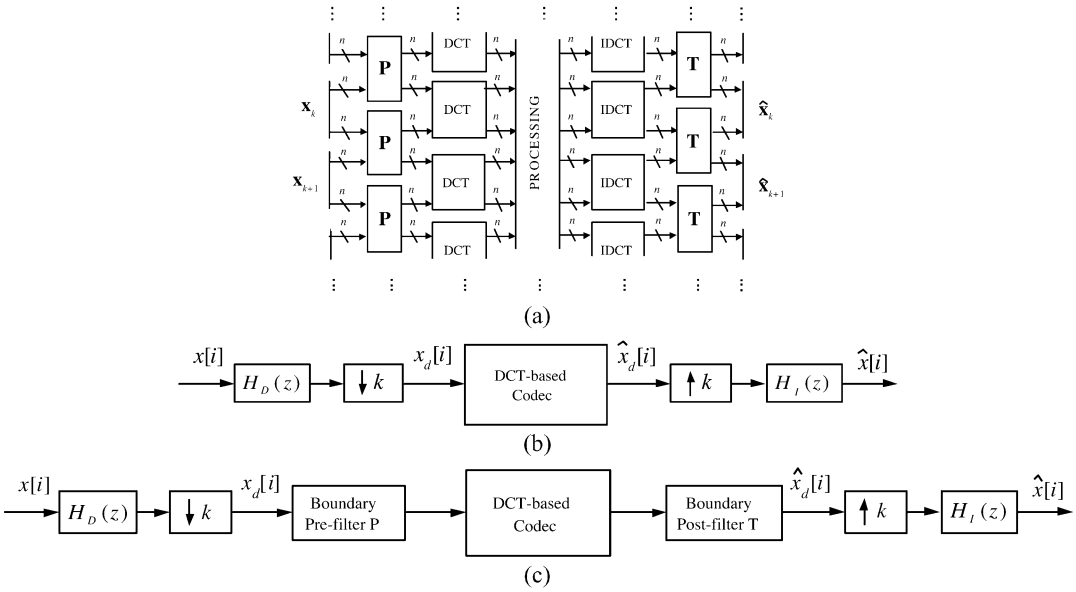


Fig. 1. Existing pre-/postfiltering schemes for DCT-based codecs. For simplicity, only the implementation diagram for 1-D signals is shown. For 2-D signals, they can be processed in a separable approach. (a) Critically sampled boundary pre-/postprocessing [7], [8], where the $N \times N$ $(N = 2n)$ matrices $\mathbf{P}$ and $\mathbf{T}$ are the prefilter and postfilter, respectively. (b) The downsampling/interpolation approach [12], [13], where $H_D(z)$ and $H_I(z)$ are the decimation and the interpolation filter, respectively, and k $(k > 1)$ is the sampling factor. (c) The cascading approach, which is a combination of (a) and (b). Here, the downsampling/interpolation operations are conducted outside the boundary pre-/postprocessing.

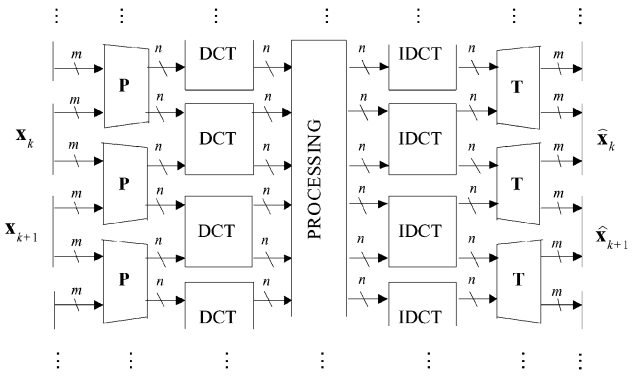


Fig. 2. Our proposed undersampled boundary pre-/postprocessing scheme for block DCT codecs. The prefilter $\mathbf{P}$ and the postfilter $\mathbf{T}$ are of sizes $N \times M$ and $M \times N$ $(M > N)$, respectively.

sampling/interpolation operations are conducted outside the boundary pre-/postprocessing. This is equivalent to replacing the DCT and the IDCT in Fig. 1(b) with the LT and the inverse LT, respectively. Instead of doing so, this paper proposes to use undersampled boundary pre-/postfilters, as shown in Fig. 2, which is an extension of Fig. 1(a) from critically sampled systems to undersampled systems. In the proposed system, the original input signal $\mathbf{x}$ is now divided into short sequences $\mathbf{x}_k$ with length of $M = 2m$ each, which is greater than the DCT's dimension $N = 2n$. Likewise, the reconstructed segment $\hat{x}_k$ is of length $M = 2m$ $(M > N)$ each. Accordingly, the sizes of the prefilter $\mathbf{P}$ and the postfilter $\mathbf{T}$ change into $N \times M$ and $M \times N$, respectively. Although M can be any integer greater than N, for simplicity, this paper only investigates the case of even M. Extension to odd M is a trivial task. Some preliminary results of undersampled systems were reported in [21]. In this paper, we will further investigate their design, implementation and applications. For image compression, simulation results show that the proposed undersampled pre-/postfiltering scheme can offer a remarkable gain over the critically sampled counterpart at high compression ratio. It also consistently outperforms the conventional downsampling/interpolation scheme in Fig. 1(b) by a large margin at various bit rates experimented. Compared with the cascading approach in Fig. 1(c), although the results at low bit rates are about the same, the proposed undersampled prefiltering/postfiltering approach has a much better performance at high bit rates. Besides, as the proposed scheme retains the merits of block-by-block processing, it is more advantageous for hardware parallel processing.

The rest of this paper is organized as follows. Section II presents the general framework of undersampled pre-/postfilters. Their relations to existing pre-/postfiltering schemes are discussed, followed by the analysis of their connections with LTs and FBs. Section III investigates the characterizations of the pre-/postfilters. $\mathbf{P}$ and $\mathbf{T}$ were constructed to possess the linear-phase property, and they are restricted to be the pseudo inverse of each other. A closed-form solution is also derived to yield minimum reconstruction error due to undersampled pre-/postprocessing. Design criteria and examples are provided in Section IV, followed by image coding results in Section V. Finally, conclusions are presented in Section VI.

Notations: Bold-faced letters indicate vectors and matrices. The following notations are used to describe submatrices of a $p \times q$ matrix $\mathbf{X}$. The jth $(1 \leq j \leq q-1)$ column is denoted as $\mathbf{X}(:, j)$. The submatrix consisting of elements in the ith through the jth columns is given by $\mathbf{X}(:, i : j)$. The submatrix consisting of elements in the i_1th through i_2th rows and j_1th through j_2th columns is represented as $\mathbf{X}(i_1 : i_2, j_1 : j_2)$. The size of a matrix is omitted when it is clear from the context.

$\mathrm{rank}(\mathbf{X})$ denotes the rank of matrix $\mathbf{X}$, i.e., the largest number of columns (or rows) of $\mathbf{X}$ that constitute a linearly independent set. $\mathrm{Trace}(\mathbf{X})$ stands for the trace of $\mathbf{X}$, i.e., the summation of its diagonal elements. For a $p \times p$ Hermitian matrix $\mathbf{X}$, denote $\lambda_i(\mathbf{X})$ (for $i = 1, \ldots, p$) as its eigenvalues, and without loss of generality, $\lambda_i(\mathbf{X})$ are sorted in an ascending order, i.e.,

$$\lambda_1(\mathbf{X}) \leq \lambda_2(\mathbf{X}) \leq \cdots \leq \lambda_{p-1}(\mathbf{X}) \leq \lambda_p(\mathbf{X}). \quad (1)$$

Special matrices are defined as follows: $\mathbf{I}_p, \mathbf{J}_p, \mathbf{0}_p$, and $\mathbf{C}_p$ denote the $p \times p$ identity matrix, the reversal matrix, the null matrix and Type-II DCT matrix,[1] respectively. Besides, $\mathbf{W}_{2p}$ represents the following butterfly matrices:

$$\mathbf{W}_{2p} = \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I}_p & \mathbf{J}_p \\ \mathbf{J}_p & -\mathbf{I}_p \end{bmatrix}. \quad (2)$$

II. Undersampled Pre-/Postfiltering Framework

A. Relations to Existing Pre-/Postfiltering Schemes

1) Connections With the Critically Sampled System: The undersampled system in Fig. 2 is an extension of the critically sampled system in Fig. 1(a), where $\mathbf{P}$ and $\mathbf{T}$ are generalized to be rectangular matrices. For an 1-D signal, the $N \times M$ $(M > N)$ prefilter $\mathbf{P}$ maps every M samples into N ones. Thus, the length of prefiltered signal is N/M of the original input. In the 2-D case, the size of the prefiltered signal is $(N/M)^2$ as the original one. The size reduction can be significant when M is large. For example, for an $L_s \times L_s$ input image, by applying an $N \times 2N$ (i.e., $M = 2N$) undersampled prefilter, the DCT block codec just needs to process $(1/4)L_s^2$ samples. Intuitively, due to the size reduction, the DCT-block codec can be much faster than that of a critically sampled one. In addition, finer quantization can be used to meet the target bit rate.

However, in the undersampled system, as $M > N$, it is impossible to achieve $\mathbf{TP} = \mathbf{I}_M$. Thus, unlike the critically sampled systems, the pre-/postfiltering operation in undersampled systems is a *lossy* process. Even without any compression (quantization), there will be some distortion in the reconstructed image. If the input signal is assumed to be uncorrelated with the quantization noise, then the total distortion (i.e., the mean square error) in the reconstructed signal can be given by

$$\sigma^2_{\mathrm{total}} = \sigma^2_r + \sigma^2_q \quad (3)$$

where σ^2_r is the distortion introduced by undersampled pre-/postfiltering, and σ^2_q is the quantization distortion. For high bit-rate coding, the undersampled pre-/postfiltering is not desirable due to the existence of σ^2_r. However, at low bit rates, with a finer quantizer, the total distortion can be less than that in the critically sampled system.

2) Relations to the Downsampling/Interpolation and the Cascading Schemes: In the downsampling/interpolation and the cascading schemes, when $k = N/M$, the length of the prefiltered signal is the same as that in the undersampled system. However, it should be emphasized that in Fig. 1(b) and (c), the size reduction is conducted through the single-input, single-output scalar filter $H_D(z)$, followed by a decimator, while in the proposed scheme of Fig. 2, it is achieved by applying a multiple input, multiple output "fat" matrix filter $\mathbf{P}$. To have a better understanding of their filtering operations, let us consider the systems with $k = 2$ in Fig. 1(b) and (c) and $M = 2N$ in Fig. 2.

[1]There are four types of DCT [1]. Type II refers to the one widely used in various image and video compression standards.

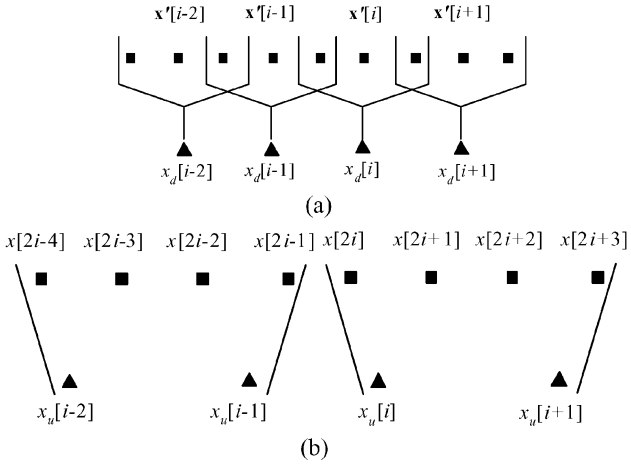


Fig. 3. Filtering process in the downsampling scheme and the proposed undersampled system. Here, "■" indicates the input samples $x[i]$. "▲" denote the output samples, where $x_d[i]$ and $x_u[i]$ represent the output of the decimator and undersampled prefilter, respectively. (a) The downsampling scheme with $k = 2$ and decimation filter length $L = 3$. The vector $\mathbf{x}'[i]$ takes the form of $\mathbf{x}'[i] = [x[2i-2]\ x[2i-1]\ x[2i]]^T$. It is an overlapped operation. (b) The undersampled system with a 2×4 (i.e., $N = 2$ and $M = 4$) prefilter. The process is a nonoverlapped block operation.

Denote the decimation filter as $H_D(z) = \sum_{n=0}^{L-1} h[n]z^n$, where L is the filter length. Suppose that the original input is $x[i]$. When $k = 2$, the decimator's output $x_d[i]$ can be written as

$$\begin{aligned} x_d[i] &= \sum_{j=0}^{L-1} h[j]x[2n-j] \\ &= [h[L-1] \quad \cdots \quad h[1] \quad h[0]]\mathbf{x}'[i] \end{aligned}$$

where $\mathbf{x}'[i]$ denotes the vector which includes all required input samples to produce $x_d[i]$, i.e.,

$$\mathbf{x}'[i] = [x[2i-L+1] \quad x[2i-L+2] \quad \cdots \quad x[2i]^T.$$

The process can be described by using a sliding window over the input signal, as shown in Fig. 3(a) (for $L = 3$). As can be seen, $\mathbf{x}'[i+1]$ can be obtained by shifting $\mathbf{x}'[i]$ by two samples. It is clear that $\mathbf{x}'[i]$ has no overlap with $\mathbf{x}'[i-1]$ when $L \leq 2$. Their overlap occurs when $L > 2$. On the other hand, for the proposed system in Fig. 2, the prefiltering stage is always a *nonoverlapped* block-by-block operation. That is, every block of M input samples is mapped into N prefiltered ones. Fig. 3(b) illustrates this process for $M = 4$ and $N = 2$.

Hence, when the filter length $L \leq 2$, Fig. 1(b) and (c) can be viewed as two special cases of Fig. 2. To be more accurate, in

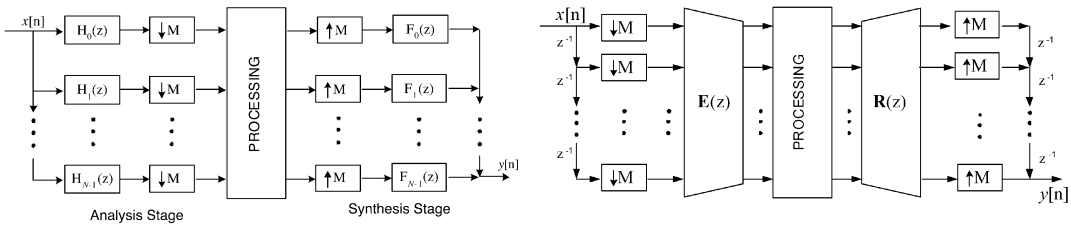


Fig. 4. N-channel undersampled filter bank with sampling factor M. (a) Direct-form implementation. (b) Polyphase implementation.

the proposed undersampled system, if the $N \times 2N$ prefilter $\mathbf{P}$ takes the form of

$$\mathbf{P} = \begin{bmatrix} h[0] & h[1] & & & & \\ & & h[0] & h[1] & & \\ & & & & \ddots & \\ & & & & h[0] & h[1] \end{bmatrix} \tag{4}$$

it is boiled down to the downsampling scheme in Fig. 1(b) with $H_D(z) = h[0] + h[1]z$ and $k = 2$, and when $\mathbf{P}$ is set to be

$$\mathbf{P} = \mathbf{P}_{\text{cr}} \begin{bmatrix} h[0] & h[1] & & & & \\ & & h[0] & h[1] & & \\ & & & & \ddots & \\ & & & & h[0] & h[1] \end{bmatrix} \tag{5}$$

it is equivalent to the cascading scheme in Fig. 1(c) with $H_D(z) = h[0] + h[1]z, k = 2$ and an $N \times N$ boundary prefilter $\mathbf{P}_{\text{cr}}$. However,, when $L > 2$, since the filtering operation in Fig. 1(b) and (c) is conducted in an overlapped manner, it cannot be represented through the undersampled system.

B. Links to Lapped Transforms and Filter Banks

Note that the system in Fig. 2 consists of two simple stages of nonoverlapped block operations, i.e., the undersampled pre-/postfiltering stage and the DCT/IDCT stage. From the transform point of view, it generates an N-band $(N = 2n), 2M$-tap $(N \times 2M)$ *undersampled LT*. Let the prefilter $\mathbf{P}$ and the postfilter $\mathbf{T}$ be partitioned as follows:

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_t \\ \mathbf{P}_b \end{bmatrix} \quad \text{and} \quad \mathbf{T} = [\, \mathbf{T}_l \quad \mathbf{T}_r \,]$$

where $\mathbf{P}_t$ and $\mathbf{P}_b$ represent $n \times M$ submatrices of $\mathbf{P}$, while $\mathbf{T}_l$ and $\mathbf{T}_r$ are $M \times n$ submatrices of $\mathbf{T}$. The corresponding forward transform matrix $\mathbf{H}$ and the backward transform matrix $\mathbf{F}$ can be expressed as [7], [8]

$$\mathbf{H} = \mathbf{C}_N \begin{bmatrix} \mathbf{P}_b & \mathbf{0}_{n \times M} \\ \mathbf{0}_{n \times M} & \mathbf{P}_t \end{bmatrix}$$

and

$$\mathbf{F} = \begin{bmatrix} \mathbf{T}_r & \mathbf{0}_{M \times n} \\ \mathbf{0}_{M \times n} & \mathbf{T}_l \end{bmatrix} \mathbf{C}_N^{-1} \tag{6}$$

respectively. Each row in $\mathbf{H}$ and each column in $\mathbf{F}$ are the basis functions. Denote the M-point input segment as $\mathbf{x}_k = \begin{bmatrix} \mathbf{x}_k^u \\ \mathbf{x}_k^d \end{bmatrix}$, where $\mathbf{x}_k^u$ and $\mathbf{x}_k^d$ contain the first m and the last m points in $\mathbf{x}_k$, respectively. The kth output of the DCT $\mathbf{c}_k$ can be computed as $\mathbf{c}_k = \mathbf{H}\mathbf{s}_k$, where the input segment $\mathbf{s}_k$ is $\mathbf{s}_k = [(\mathbf{x}_{k-1}^d)^T \quad (\mathbf{x}_k)^T \quad (\mathbf{x}_{k+1}^u)^T]^T$. The term "lapped" is so called as $\mathbf{s}_k$ has an overlap of M samples with its adjacent neighbors $\mathbf{s}_{k-1}$ and $\mathbf{s}_{k+1}$ each. Accordingly, at the output, the reconstructed segment $\hat{\mathbf{x}}_k$ can be recovered from $\hat{\mathbf{c}}_k$ in an overlap-add fashion [7], [8], [18]–[20]. The term "undersampled" implies that the total number of DCT coefficients $\mathbf{c}_k$ is less than that of the original input $\mathbf{x}_k$. Note that it has been widely known that by replacing the DCT with the LT, the blocking artifacts can be effectively reduced [18]–[20]. With the extension to undersampled systems, we aim to get even better performances for low bit-rate coding.

From the perspective of multirate signal processing, the undersampled LT is linked to an N-channel undersampled FB with the sampling factor of M, as shown in Fig. 4(a). The analysis filters $H_i(z)$ (for $i = 1, \ldots, N$) and synthesis filters $F_i(z)$ (for $i = 1, \ldots, N$) are of length $L = 2M$ each, which can be written into

$$\begin{aligned} &[\, H_1(z) \quad H_2(z) \quad \cdots \quad H_N(z) \,]^T \\ &\quad = \mathbf{H}[\, 1 \quad z \quad \cdots \quad z^M \,]^T; \\ &[\, F_1(z) \quad F_2(z) \quad \cdots \quad F_N(z) \,] \\ &\quad = [\, z^M \quad z^{M-1} \quad \cdots \quad 1 \,]\mathbf{F}. \end{aligned} \tag{7}$$

Here, $H_1(z)$ and $F_1(z)$ are low-pass filters and the remaining ones are band-pass/high-pass filters. An equivalent, but alternative, form of Fig. 4(a) is through the polyphase representation in Fig. 4(b), where the analysis polyphase matrix $\mathbf{E}(z)$ and the synthesis polyphase matrix $\mathbf{R}(z)$ can be written through the transform matrix $\mathbf{H}$ and the inverse transform matrix $\mathbf{F}$ as follows:

$$\mathbf{E}(z) = \mathbf{H} \begin{bmatrix} \mathbf{I}_m \\ z\mathbf{I}_m \end{bmatrix} \quad \mathbf{R}(z) = [\, z\mathbf{I}_m \quad \mathbf{I}_m \,]\, \mathbf{F}. \tag{8}$$

By the definitions of $\mathbf{H}$ and $\mathbf{F}$ in (6), the explicit expressions of $\mathbf{E}(z)$ and $\mathbf{R}(z)$ are

$$\mathbf{E}(z) = \mathbf{C}_N \hat{\Lambda}_N(z)\mathbf{P}, \quad \mathbf{R}(z) = \mathbf{T}\hat{\Lambda}_N^{-1}(z)\mathbf{C}_N^T \tag{9}$$

in which $\mathbf{C}_N$ is the Type-II DCT matrix and $\hat{\Lambda}_N(z) = \begin{bmatrix} \mathbf{0}_n & \mathbf{I}_n \\ z\mathbf{I}_n & \mathbf{0}_n \end{bmatrix}$. In the polyphase implementation, as the input signal is first downsampled, the processing speed is much faster than that in the direct-form implementation. In addition, one can easily get a VLSI-friendly hardware implementation structure through the polyphase representation [22].

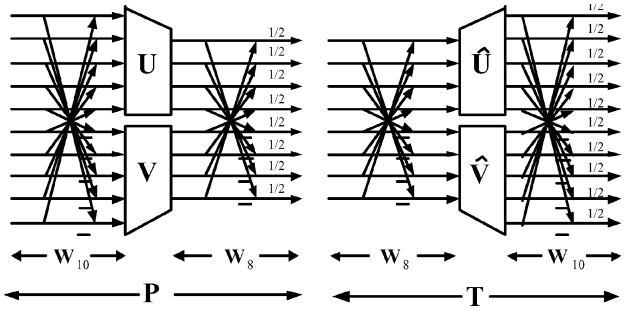


Fig. 5. Structures of the prefilter and the postfilter with the linear-phase property ($N = 8$ and $M = 10$). Here, $\mathbf{W}_8$ and $\mathbf{W}_{10}$ are butterfly matrices defined as in (2). $\{\mathbf{U}, \mathbf{V}\}$ are 4×5 matrices while $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ and 5×4 ones.

III. Expressions of Pre-/Postfilters

This section derives the expressions of $\mathbf{P}$ and $\mathbf{T}$. Just as in critically sampled systems, the linear-phase and the (pseudo) inverse conditions are imposed. As we have mentioned before, the undersampled pre-/postfiltering is an *irreversible* process. Even without any quantization, it will lead to the reconstruction error σ_r^2, as shown in (3). To further simplify the design, a closed-form solution is derived to minimize the reconstruction error σ_r^2.

A. Linear-Phase Property

The linear-phase property is crucial in many applications of image processing. Not only does it lead to reconstructed image and video signals with good visual quality, it also offers an effective representation of finite-length signals via symmetric extensions at signal boundaries [8], [18].

For the undersampled LT in Fig. 2, the linear-phase property implies that the basis functions of $\mathbf{H}$ and $\mathbf{F}$ are either symmetric or antisymmetric. Following the similar derivations of critically sampled systems in [7] and [18], it can be shown that the linear-phase property holds if and only if $\mathbf{P}$ and $\mathbf{T}$ take the following forms:

$$\mathbf{P} = \mathbf{W}_N \begin{bmatrix} \mathbf{U} & \mathbf{0} \\ \mathbf{0} & \mathbf{V} \end{bmatrix} \mathbf{W}_M$$

and

$$\mathbf{T} = \mathbf{W}_M \begin{bmatrix} \hat{\mathbf{U}} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{V}} \end{bmatrix} \mathbf{W}_N \tag{10}$$

where $\mathbf{W}_N$ and $\mathbf{W}_M$ are butterfly matrices as in (2), $\{\mathbf{U}, \mathbf{V}\}$ are $n \times m$ free matrices, and $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ are $m \times n$ free matrices. With (10), the odd-numbered filters $H_{2i-1}(z)$ and $F_{2i-1}(z)$ (for $i = 1, \ldots, n$) are symmetric, while the even-numbered ones $H_{2i}(z)$ and $F_{2i}(z)$ are anti-symmetric. As an example, Fig. 5 shows the detailed implementations of $\mathbf{P}$ and $\mathbf{T}$ for $N = 8, M = 10$. It can be seen that in the preprocesing stage, the ten input samples first go through the butterfly matrix $\mathbf{W}_{10}$. Two small 4×5 matrices $\mathbf{U}$ and $\mathbf{V}$ then map the first and the last five samples into four ones each, yielding eight samples in total. These eight samples are filtered by another butterfly matrix $\mathbf{W}_8$ to produce the preprocessed samples. The postprocessing stage follows nearly the same path.

B. The Pseudo Inverse Restriction

Recall that, in critically sampled systems with $N = M$ [7], [18], besides the linear-phase property, the perfect reconstruction condition is often imposed, in which $\mathbf{P}$ and $\mathbf{T}$ are the exact inverse of each other, i.e.,

$$\mathbf{P} = \mathbf{T}^{-1}. \tag{11}$$

In undersampled systems, as the prefilter $\mathbf{P}$ is a "fat" matrix, it has no left-inverse. Hence, there exists no postfilter $\mathbf{T}$ which meets the perfect reconstruction property. To simplify the design, we restrict $\mathbf{P}$ and $\mathbf{T}$ to be the *pseudo-inverse* [23] of each other, i.e.,

$$\mathbf{P} = \mathbf{T}^{+}. \tag{12}$$

Note that (12) is an extension of (11) to undersampled systems. In the special case of $N = M$, (12) is equivalent to (11). Under this constraint, for a given prefilter $\mathbf{P}$, one can get the optimal solution of $\mathbf{TPx} = \mathbf{x}$ in the least square sense [24]. Using the orthogonality of $\mathbf{W}_N$ and $\mathbf{W}_M$, we know that (12) is equivalent to

$$\mathbf{U} = \hat{\mathbf{U}}^{+} \quad \text{and} \quad \mathbf{V} = \hat{\mathbf{V}}^{+} \tag{13}$$

i.e., $\mathbf{U}$ and $\mathbf{V}$ are the pseudo inverses of $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, respectively. Due to the uniqueness of the pseudo inverse, we just need to design $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$, while $\mathbf{U}$ and $\mathbf{V}$ can be determined by (13). Obviously, the maximum degrees of design freedom can be gained when $\mathbf{U}$ and $\mathbf{V}$ are of full rank, i.e.,

$$\operatorname{rank}(\mathbf{U}) = \operatorname{rank}(\mathbf{V}) = \mathbf{n}. \tag{14}$$

Hence, in what follows, we shall only consider this case.

C. Minimization of the Reconstruction Error

From (10) and (13), it is clear that by imposing the linear-phase and the pseudo inverse conditions, the design and optimization of $\mathbf{P}$ and $\mathbf{T}$ can be converted into that of two $m \times n$ matrices $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$. The total number of free parameters in these two matrices is 2 nm. For a large number of channels $N = 2n$ and a large sampling factor $M = 2m$, this number can be fairly large, which may slow down the convergence in the pre-/postfilter optimization. It would be useful to cut down the number of free parameters by structurally imposing other constraints [8], [18]–[20].

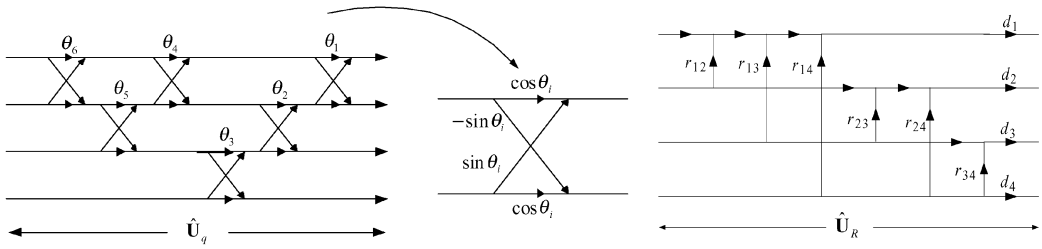
Recall that, in critically sampled systems, under the perfect reconstruction condition in (11), the reconstruction error is zero if the quantization stage is omitted. For undersampled systems, we will further restrict $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ so that the reconstruction error σ_r^2 in (3) is minimized. Recall that σ_r^2 is the MSE of reconstructed signal due to undersampled pre-/postprocessing. It is easy to show that σ_r^2 can be written as

$$\begin{aligned} \sigma_r^2 &= \frac{1}{M} E\left\{\|(\mathbf{I} - \mathbf{TP})\mathbf{x}\|_2^2\right\} \\ &= \frac{1}{M} \operatorname{Trace}(\mathbf{E}\mathcal{R}_{xx}\mathbf{E}^T) \end{aligned} \tag{15}$$

where $\mathbf{E} = \mathbf{I} - \mathbf{TP}$ and $\mathcal{R}_{xx}$ is the autocorrelation matrix of the input signal $\mathcal{R}_{xx} = E\{\mathbf{x}\mathbf{x}^T\}$. In what follows, we aim to exploit a couple of questions. 1) Given N, M, and $\mathcal{R}_{xx}$, what is the minimal value of σ_r^2? 2) How do we characterize the free matrices $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ so that σ_r^2 is minimized?

Detailed derivations of the above-mentioned questions are given in the Appendix. Here, we just outline the results. Let $\mathcal{R}'_{xx} = \mathbf{W}_M \mathcal{R}_{xx} \mathbf{W}_M$, and define $\mathcal{R}_u$ and $\mathcal{R}_v$ as the $m \times m$ upper-left and lower-right submatrices of $\mathcal{R}'_{xx}$, i.e.,

$$\mathcal{R}_u = \mathcal{R}'_{xx}(1:m, 1:m) \tag{16}$$



(a) (b)

Fig. 6. Detailed parameterizations of $\hat{\mathbf{U}}_q$ and $\hat{\mathbf{U}}_R$ when $n = 4$). (a) Parameterization of $\hat{\mathbf{U}}_q$, where the rotation angles θ_i (for $i = 1, \ldots, 6$) are free parameters. (b) Parameterization of $\hat{\mathbf{U}}_R$, where the scaling factors r_i (for $i = 1, \ldots, 4$) and the lifting coefficients r_{ij} (for $i = 1, \ldots, 4$ and $j = i, \ldots, 4$) are free parameters.

TABLE I
VALUES OF $\sigma^2_{r,\min}$ BY USING THE UNDERSAMPLED LTS WITH $N = 8$

M	8	10	12	14	16
$\sigma^2_{r,min}$	0	0.0055	0.0098	0.0136	0.0171

$$\mathcal{R}_v = \mathcal{R}'_{xx}(m+1:M, m+1:M). \tag{17}$$

As $\mathcal{R}_{xx}$ is a Hermitian matrix, it is obvious that both $\mathcal{R}_u$ and $\mathcal{R}_v$ are also Hermitian. Hence, through the eigen-decomposition [23], they can be factorized into

$$\mathcal{R}_u = \mathbf{A}_u \Lambda_u \mathbf{A}_u^T, \quad \mathcal{R}_v = \mathbf{A}_v \Lambda_v \mathbf{A}_v^T \tag{18}$$

where $\mathbf{A}_u$ and $\mathbf{A}_v$ are $m \times m$ orthogonal matrices; and Λ_u and Λ_v are diagonal matrices with diagonal entries $\Lambda_u(i,i) = \lambda_i(\mathcal{R}_u)$ and $\Lambda_v(i,i) = \lambda_i(\mathcal{R}_v)$ (for $i = 1, \ldots, m$), respectively.

It is proved in the Appendix that the minimal value of σ_r^2 is

$$\sigma^2_{r,\min} = \min(\sigma_r^2) = \frac{1}{M}\sum_{i=1}^{m-n} \lambda_i(\mathcal{R}_u) + \frac{1}{M}\sum_{i=1}^{m-n} \lambda_i(\mathcal{R}_v). \tag{19}$$

That is, $\sigma^2_{r,\min}$ is the sum of $m-n$ smallest eigenvalues of $\mathcal{R}_u$ and that of $\mathcal{R}_v$. As an example, for $N = 8$ ($n = 4$), Table I lists the value of $\sigma^2_{r,\min}$ with different values of M. Here, the input signal is assumed to be AR(1) with the correlation coefficient of $\rho = 0.95$, i.e., $\mathcal{R}_{xx}(i,j) = 0.95^{i-j}$. Just as expected, $\sigma^2_{r,\min}$ increases when M becomes large. And when $M \to \infty, \sigma^2_{r,\min} = \sigma_x^2 = 1$, which indicates that the whole signal is lost.

As shown in the Appendix, $\sigma^2_{r,\min}$ is reached if and only if $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ are characterized as

$$\begin{aligned} \hat{\mathbf{U}} &= \mathbf{A}_u(:, m-n+1:m)\hat{\mathbf{U}}_q\hat{\mathbf{U}}_R \\ \hat{\mathbf{V}} &= \mathbf{A}_v(:, m-n+1:m)\hat{\mathbf{V}}_q\hat{\mathbf{V}}_R \end{aligned} \tag{20}$$

where $\mathbf{A}_u(:, m-n+1:m)$ and $\mathbf{A}_v(:, m-n+1:m)$ denote the last n columns of $\mathbf{A}_u$ and $\mathbf{A}_v$, respectively. $\{\hat{\mathbf{U}}_R, \hat{\mathbf{V}}_R\}$ are arbitrary $n \times n$ upper triangular matrices with nonzero diagonal entries and $\{\hat{\mathbf{U}}_q, \hat{\mathbf{V}}_q\}$ are arbitrary $n \times n$ orthonormal matrices.

Note that, for a given $\mathcal{R}_{xx}$, $\mathbf{A}_u$ and $\mathbf{A}_v$ can be determined by (18). Hence, in (20), we just need to parameterize two $n \times n$ orthogonal matrices $\hat{\mathbf{U}}_q$ and $\hat{\mathbf{V}}_q$ and two $n \times n$ up-triangular matrices $\hat{\mathbf{U}}_R$ and $\hat{\mathbf{V}}_R$. Recall that an $n \times n$ orthogonal matrix can be completely parameterized through $(n(n-1))/(2)$ Givens rotations [22], while an $n \times n$ up-triangular matrix can be always represented by n diagonal elements and $(n(n-1))/(2)$ lifting steps [25]. Therefore, to minimize σ_r^2, the total number of free parameters is $2n^2 = N^2/2$, which is regardless of the sampling factor M. As we have said, $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ contain 2-nm free parameters. Under the restriction of (19), we gain a reduction of $2n(m-n)$ parameters, which is advantageous in fast optimization. For illustration purpose, when $n = 4$, Fig. 6(a) and (b) shows the detailed parameterizations of $\hat{\mathbf{U}}_q$ and $\hat{\mathbf{U}}_R$, respectively. Here, the rotation angles θ_i (for $i = 1, \ldots, 6$) are free parameters in $\hat{\mathbf{U}}_q$, while the scaling factors d_i (for $i = 1, \ldots, 4$) and the lifting coefficients r_{ij} (for $i = 1, \ldots, 4$ and $j = i, \ldots, 4$) are those in $\hat{\mathbf{U}}_R$. They can vary freely and independently under a mild constraint that $d_i \neq 0$ (for $i = 1, \ldots, 4$).

IV. DESIGN CRITERIA AND EXAMPLES

A. Design Criteria

As our target application is low bit-rate coding, the cost function C_{cost} we choose is a linear combination of the DC leakage C_{dc} and the coding gain C_{cg}, both of which are crucial to compression [26]

$$C_{\text{cost}} = \alpha_1 C_{\text{dc}} + \alpha_2 C_{\text{cg}} \tag{21}$$

where α_1 and α_2 are the weighting factors. Through empirical studies, we found out that $\alpha_1 = 1$ and $\alpha_2 = -10$ can yield good compression performance.

The DC leakage C_{dc} in (21) measures the amount of energy that leaks into highpass subbands when the input signal is a unit constant. Low DC leakage is effective in reducing the blocking and checkerboard artifacts [26], [27]. C_{dc} can be expressed through the analysis filters $H_i(z)$ as

$$C_{\text{dc}} = \sum_{i=1}^{n-1} |H_{2i+1}(1)|. \tag{22}$$

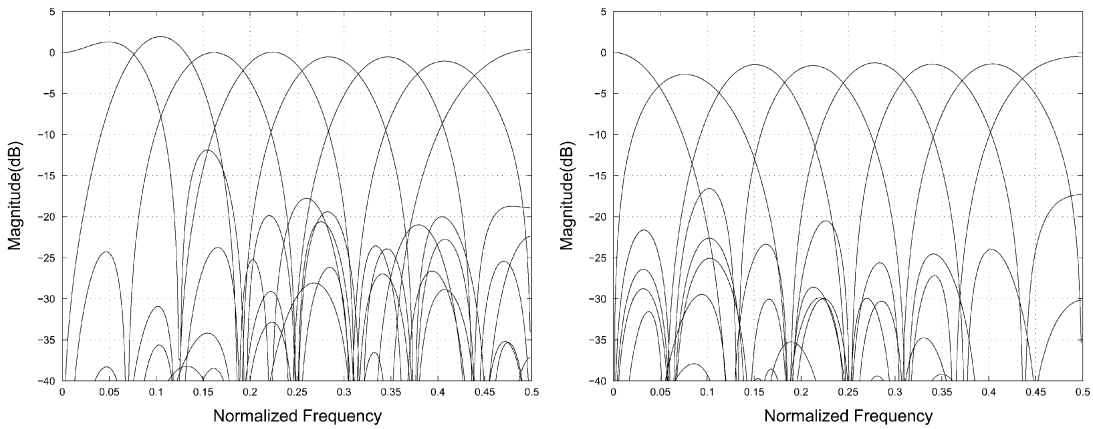
Note that only odd-numbered filters $H_{2i+1}(z)$ (for $i = 1, \ldots, n-1$) are considered in (22) since the even-numbered ones $H_{2i}(z)$ are antisymmetric, which have zero frequency responses at $z = 1$, i.e., $H_{2i}(1) = 0$.

The coding gain C_{cg} is defined as the ratio of the quantization distortion σ^2_{PCM} using straight PCM quantization to the total distortion σ^2_{total} in transform coding

$$C_{\text{cg}} = 10 \log_{10} \frac{\sigma^2_{\text{PCM}}}{\sigma^2_{\text{total}}} \tag{23}$$

TABLE II
VALUES OF FREE MATRICES IN UNDERSAMPLED LTS WITH $N = 8$

Transforms	M	$\hat{\mathbf{U}}$	$\hat{\mathbf{V}}$
LT-88	8	$\mathbf{I}_4$	$\begin{bmatrix} 0.7029 & -0.5740 & 0.0906 & -0.0575 \\ 0.3905 & 0.5838 & -0.4625 & -0.0423 \\ 0.0510 & 0.2515 & 0.7243 & -0.2981 \\ 0.0199 & 0.0071 & 0.0783 & 0.8107 \end{bmatrix}$
LT-810	10	$\begin{bmatrix} 1.0261 & -0.0720 & 0.0391 & -0.0069 \\ 0.2347 & 0.8550 & -0.1425 & 0.0411 \\ -0.0453 & 0.5410 & 0.5766 & -0.0873 \\ 0.1162 & -0.1911 & 0.8297 & 0.2334 \\ 0.0434 & 0.0664 & -0.1255 & 1.0021 \end{bmatrix}$	$\begin{bmatrix} 0.5465 & -0.4728 & 0.1173 & -0.0642 \\ 0.6240 & 0.0787 & -0.3308 & 0.0040 \\ 0.0752 & 0.7401 & -0.0471 & -0.1772 \\ 0.0839 & -0.0135 & 0.8215 & -0.1233 \\ 0.0079 & 0.0457 & -0.0220 & 0.8566 \end{bmatrix}$
LT-816	16	$\begin{bmatrix} 1.0713 & -0.2138 & 0.0548 & -0.0097 \\ 0.6638 & 0.3515 & -0.1353 & 0.0341 \\ 0.1550 & 0.8656 & -0.1334 & 0.0212 \\ -0.0885 & 0.7756 & 0.2881 & -0.0715 \\ -0.0350 & 0.2194 & 0.8043 & -0.0823 \\ 0.0685 & -0.1917 & 0.8546 & 0.1787 \\ 0.0365 & -0.1205 & 0.3253 & 0.6677 \\ -0.0619 & 0.1463 & -0.2474 & 1.0684 \end{bmatrix}$	$\begin{bmatrix} 0.0456 & 0.3376 & -0.3926 & 0.0885 \\ 0.2872 & 0.5244 & -0.7247 & 0.1579 \\ 0.6428 & 0.0302 & -0.2679 & 0.0065 \\ 0.6648 & -0.2544 & 0.3879 & -0.2501 \\ 0.2106 & 0.1921 & 0.5765 & -0.3334 \\ -0.2249 & 0.6716 & 0.3344 & -0.0479 \\ -0.1799 & 0.3992 & 0.1117 & 0.4931 \\ 0.1090 & -0.2127 & 0.0720 & 0.9073 \end{bmatrix}$



(a) (b)

Fig. 7. Frequency responses of the LT-88 ($N = 8, M = 8$) [18]. (a) Analysis bank. (b) Synthesis bank.

Suppose that an input signal x with variance σ_x^2 is coded at B bits per sample. Then, σ_{PCM}^2 is expressed as [22]

$$\sigma_{\mathrm{PCM}}^2 = c2^{-2B}\sigma_x^2, \tag{24}$$

where c is some constant depending on the characteristics of $\mathbf{x}$. According to [22], if $\mathbf{x}$ is a Gaussian signal with zero-mean, a good approximation is $c = 0.75$. For the undersampled LT studied in this paper, the total distortion $\sigma_{\mathrm{total}}^2$ is given by (3). σ_r^2 is the mean square error when the quantization stage is omitted, as expressed in (15), with its minimal value $\sigma_{r,\min}^2$ given in (19). σ_q^2 represents the quantization error, which takes the form of

$$\sigma_q^2 = \frac{c}{M}\sum_{i=1}^{N} 2^{-2B_i}\sigma_i^2 \parallel \mathbf{F}(:,i) \parallel^2 \tag{25}$$

in which B_i ($B_i \geq 0$) are the number of bits allocated to the ith subbband, with their sum equals MB, i.e.,

$$\sum_{i=1}^{N} B_i = MB. \tag{26}$$

σ_i^2 is the variance of the ith subband signal and $\mathbf{F}(:,i)$ is the ith column of $\mathbf{F}$, or the ith synthesis filter. In (25), it is assumed that the quantization noise is white and uniformly distributed [27], [28]. At high bit rates (i.e., large B), (25) can be simplified to [28]

$$\sigma_q^2 = \frac{cN2^{\frac{-2BM}{N}}}{M}\left(\prod_{i=0}^{N-1}\sigma_i^2\|\mathbf{F}(:,i)\|\right)^{\frac{1}{N}}. \tag{27}$$

At low bit rates, the closed-form solution of σ_q^2 in (25) is still unknown and it can be calculated through the nonlinear optimiza-

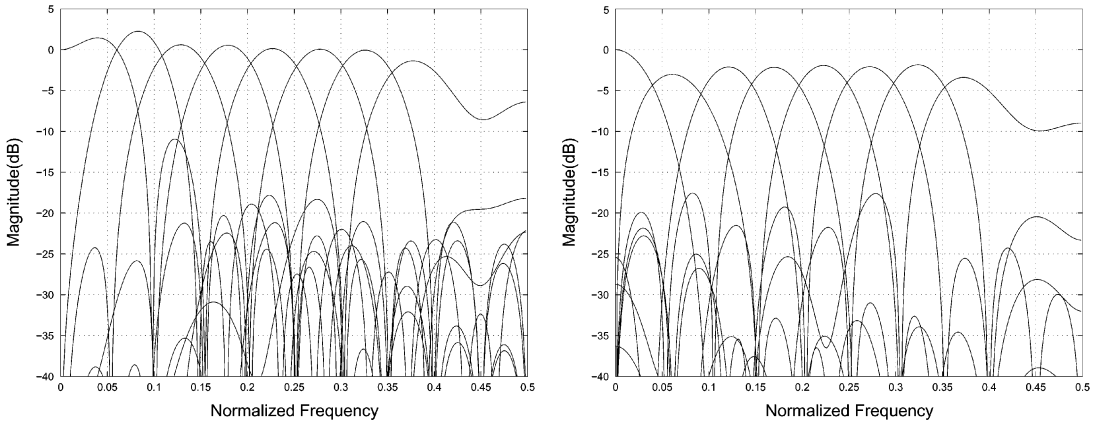


Fig. 8. Frequency responses of the LT-810 ($N = 8, M = 10$). (a) Analysis bank. (b) Synthesis bank.

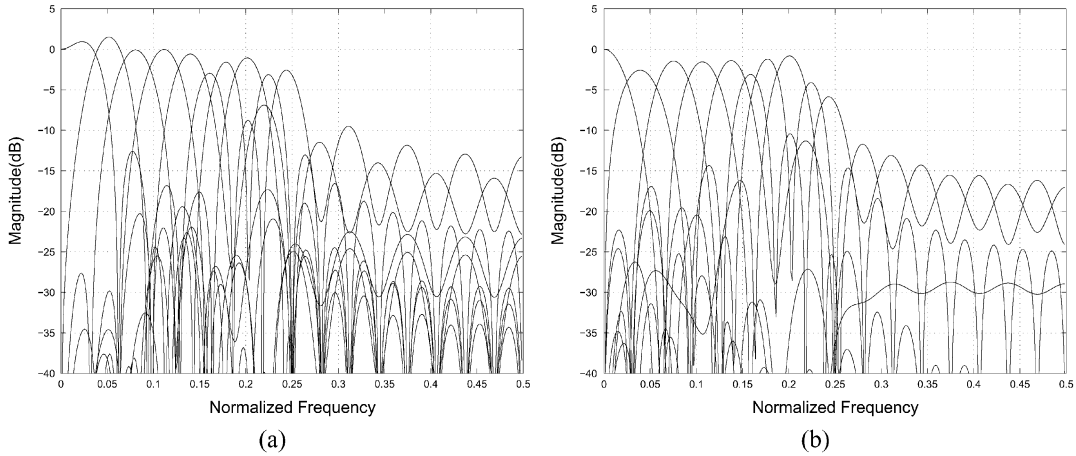


Fig. 9. Frequency responses of the LT-816 ($N = 8, M = 16$). (a) Analysis bank. (b) Synthesis bank.

tion [20]. Since the optimization of an LT (or FB) usually involves thousands of iteration steps, for simplicity, we use (27), rather than (25) during the optimization. According to [20] and [28], (27) works well for $B \geq 1$ bpp. As we are interested in low bit-rate coding, the bit rate B is fixed to be $B = 1$ bpp in our design. Following the convention of filter bank optimization, the input signal $\mathbf{x}$ in use is an AR(1) process with the correlation coefficient $\rho = 0.95$, i.e., $\mathcal{R}_{xx}(i,j) = 0.95^{i-j}$ [18]–[20]. Besides, $\mathbf{x}$ is assumed to be Gaussian [20] with $c = 0.75$ in (24) and (27) [22].

B. Examples

The optimization of (21) is carried out through the unconstrained nonlinear optimization routine *fminunc* in Matlab. The free matrices $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ are parameterized to yield $\sigma_{r,\min}$, as discussed in Section IV. During the design, $\hat{\mathbf{U}}_q$, $\hat{\mathbf{V}}_q$, $\hat{\mathbf{U}}_R$, and $\hat{\mathbf{V}}_R$ are initialized to be the identity matrix $\mathbf{I}$. That is, in Fig. 6, the rotation angles θ_i are initialized to zeros, while the diagonal multipliers d_i and r_{ij} are all initialized to ones. As 8×8 DCT is widely used in JPEG and MPEG standards, we present two design examples LT-810 and LT-816 for $N = 8$. Here, the notation LT-$8M$ denotes the LT with an $8 \times M$ prefilter. The free matrices of $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ of LT-810 and LT-816 are tabulated in Table II. For comparison, the free matrices of the critically sampled LT-88 [20] are also presented. This LT has the highest coding gain at high bit rates ($C_{\text{cg}} = 9.61$ dB) [20].

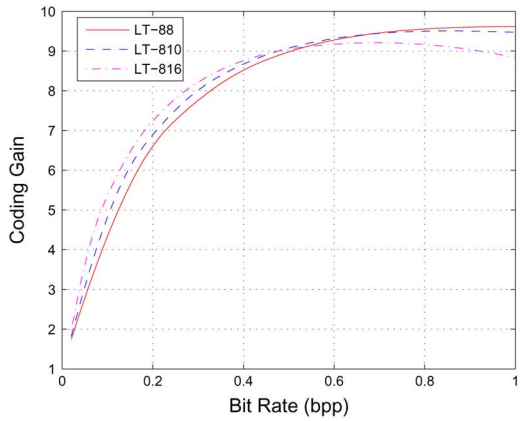


Fig. 10. Comparison of theoretical coding gains at low bit rates for different LTs.

Figs. 7–9 show the frequency responses of three LTs. As can be seen, with the increase of the sampling factor M, the passbands are narrower at low frequency bands, while the frequency resolutions become worse at high frequency bands. This is understandable as more high frequency components would be discarded when M gets larger.

Fig. 10 further compares the theoretical coding gain of different LTs at low bit rates, where the calculations follow the guidelines in [20]. One can observe that at $B < 0.5$ bpp, the LT-810 and the

TABLE III
OBJECTIVE CODING PERFORMANCE (PSNR IN DECIBELS)

Images	bpp	JPEG	JPEG2000	LT-88	Downsampling/Interpolation		Cascading Scheme		LT-810	LT-816
					$k = 1.25$	$k = 2$	$k = 1.25$	$k = 2$		
Lena	0.03125	-	25.22	25.15	25.05	25.58	25.77	**26.23**	25.53	**26.23**
	0.05	-	27.04	27.23	26.63	27.19	27.58	27.77	27.37	**27.78**
	0.1	-	29.87	30.14	29.47	29.43	**30.28**	29.78	30.24	30.19
	0.16	27.32	31.98	32.25	31.23	30.67	32.00	30.90	**32.35**	31.92
	0.25	30.76	34.10	34.31	32.64	31.46	33.19	31.60	**34.35**	33.36
Peppers	0.03125	-	24.59	23.98	23.78	24.51	24.95	**25.41**	25.26	25.20
	0.05	-	26.38	26.43	25.87	26.30	27.02	27.10	**27.34**	26.99
	0.1	-	29.73	29.78	28.89	28.58	29.83	29.24	**30.32**	29.29
	0.16	26.04	31.68	31.76	30.66	29.72	31.34	30.29	**32.04**	30.64
	0.25	30.18	33.53	33.47	32.06	30.44	32.52	30.94	**33.56**	31.64
Boat	0.03125	-	23.81	23.32	23.22	23.88	23.65	**24.29**	23.45	24.22
	0.05	-	25.04	24.64	24.84	25.02	25.24	25.47	25.06	**25.50**
	0.1	-	27.32	27.31	27.13	26.60	**27.50**	27.21	27.42	**27.50**
	0.16	25.10	29.04	29.27	28.70	27.58	29.07	28.22	**29.36**	28.92
	0.25	28.43	31.03	31.34	30.09	28.29	30.53	28.94	**31.35**	30.20
Goldhill	0.03125	-	24.95	24.63	24.55	24.75	25.07	**25.56**	24.92	25.52
	0.05	-	26.05	26.03	25.77	25.73	26.23	26.55	26.22	**26.58**
	0.1	-	27.79	27.93	27.66	27.27	28.07	27.60	28.02	**28.16**
	0.16	26.05	29.21	29.34	28.85	28.12	29.27	28.31	**29.37**	29.22
	0.25	28.34	30.54	**30.81**	29.94	28.80	30.35	28.87	30.72	30.24
Barbara	0.03125	-	21.88	20.99	21.56	21.90	22.18	22.39	22.05	**22.43**
	0.05	-	22.79	22.96	22.53	22.71	23.25	23.09	**23.40**	23.16
	0.1	-	24.65	25.43	24.13	23.62	24.82	23.83	**25.82**	24.14
	0.16	-	26.25	27.45	25.09	24.07	25.74	24.20	**27.54**	24.71
	0.25	24.38	28.36	**29.63**	25.98	24.37	26.42	24.44	29.19	25.15

Here, parameter k is the downsampling factor, and the symbol "-" indicates that the bitrate is not achievable in JPEG even when the quality factor is set to zero.

LT-816 can provide higher coding gain than the LT-88, which indicates the promising applications of undersampled systems in low bit-rate coding. When the bit rate gets higher, the coding gain of the LT-816 drops quickly, but that of the LT-810 is still very close to the LT-88. It should be also noted that the theoretical coding gains in Fig. 10 do not take into account entropy coding, which often exists in practical image coding systems. With Huffman/arithmetic coding, the actual bit rates would be much lower than those in Fig. 10, as we will demonstrate in Section V.

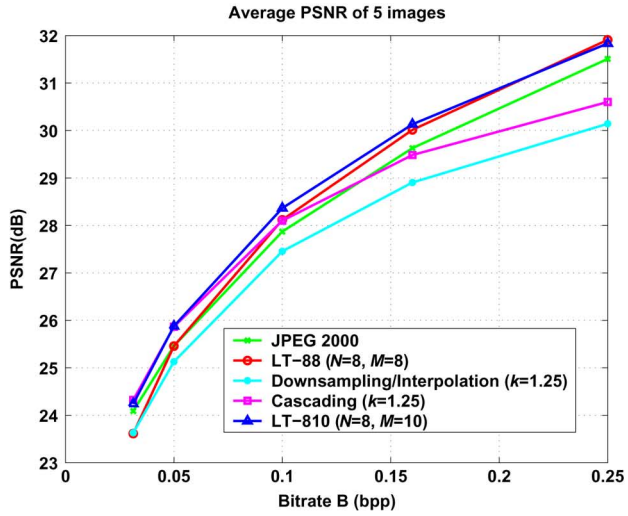
V. LOW BIT-RATE IMAGE CODING RESULTS

A. Experiment Configurations

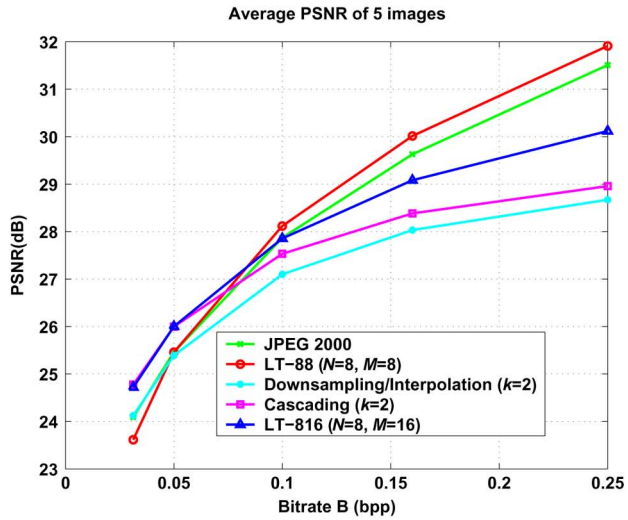
In this section, we present low bit-rate image compression results of block DCT coders using different pre-/postfiltering operators. Specifically, the pre-/postfiltering schemes under comparison include the following.

- The critically sampled boundary pre-/postfilters in Fig. 1(a), where the free matrices $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ are chosen to be those of the LT-88 in Table II.
- The downsampling/interpolation scheme of Fig. 1(b). Just as in [12], the downsampling/interpolation operations are implemented through the *imresize* function in Matlab 6.0. To have a fair comparison with the LT-810 and the LT-816, the downsampling factor k is chosen to be $k = 10/8 = 1.25$ and $k = 16/8 = 2$, respectively.
- The cascading scheme of Fig. 1(c). Here, $H_D(z)$ and $H_I(z)$ and k are the same as those in the downsampling/interpolation scheme and the boundary pre-/postfilters are identical to those of the LT-88.
- The proposed undersampled pre-/postfilters LT-810 and LT-16, whose free matrices $\{\hat{\mathbf{U}}, \hat{\mathbf{V}}\}$ are listed in Table II.

A JPEG-like codec, called as the L-CEB [14], is employed to quantize and code the DCT-coefficients. This codec can provide comparable compression performances to the state-of-the-art compression standard JPEG2000 [16] at much lower implementation cost. It differs from the JPEG in that it is based on more advanced context-based entropy coding, rather than the run-length and Huffman coding. Five test images with different flavors are used: *Lena, Peppers, Boat, Goldhill*, and *Barbara*, all of which are 8-bit grayscale 512×512 images.



(a)



(b)

Fig. 11. Average PSNR compression results of five images (*Lena, Peppers, Boat, Goldhill, Barbara*, using the L-CEB codec. In (a), results of the downsampling/interpolation scheme and the cascading scheme with $k = 1.25$ are compared with our proposed LT-810 as they have nearly the same coding complexities. Likewise, in (b), results of the downsampling/interpolation scheme and the cascading scheme with $k = 2$ are compared with our proposed LT-816. In both (a) and (b), results obtained by the LT-88 and JPEG2000 are included to serve as the benchmark.

Table III provides the detailed objective PSNR results for bit rates ranging from 0.03125 to 0.25 bpp; i.e., ranging from 256:1 to 32:1 in terms of compression ratio (CR). Results from the JPEG and the JPEG2000 are also included to serve as benchmarks. For the JPEG results in Table III, "-" implies that the corresponding bit rate cannot be achieved even when the quality factor is set to 0. The JPEG2000 implementation is based on the Kakadu software [29] (Version 5.1.1) in the single layer (SL) mode. The JPEG2000-SL is optimized for rate-distortion performance, but it is not scalable. To be fair, the 114-byte header in the JPEG2000 is excluded in counting the bit rates. In each row of Table III, the best result is highlighted in boldface. The average PSNR results of these five images are further presented in Fig. 11 To have fair comparisons, for the downsampling/interpolation and the cascading schemes, results of $k = 1.25$ and $k = 2$ are grouped together with those of the LT-810 and the LT-816, respectively. Reconstructed images of *Barbara* are demonstrated in Figs. 12 and 13 for 0.03125 and 0.16 bpp, respectively.

B. Comparison With Critically Sampled Schemes

This subsection compares the performances of the proposed undersampled LTs (i.e., the LT-810 and the LT-816) with those obtained by the critically sampled counterpart LT-88.

Table III indicates that the LT-810 outperforms the LT-88 in most cases with the PSNR gain up to 1.28 dB (in the case of *Peppers* coded at 0.03125 bpp). The average improvement is around 0.6 dB at $B = 0.3125$ bpp and with less improvement around 0.1 dB at 0.16 bpp. At 0.25 bpp, the LT-810 is just slightly worse than the LT-88. The visual qualities of reconstructed images conform with the objective PSNR results. For a 2-D image, the number of prefiltered samples generated in the LT-810 is only $((8/10))^2 \times 100\% = 64\%$ as that in the LT-88. Accordingly, the complexity of the DCT-block coders is much less than that of the LT-88.

The LT-816 can yield higher PSNR than the LT-88 at very low bit rates with a gain of 0.9 dB to 1.48 dB at 0.03125 bpp $(\mathrm{CR} = 256 : 1)$ and around 0.5 dB on average at at 0.05 bpp $(\mathrm{CR} = 160 : 1)$. At these bit rates, the superiority of visual quality is also obvious, as testified by Fig. 12. The LT-816 can substantially reduce the blocking artifacts exhibited in the LT-88. However, when the bit rate goes higher, the LT-816 is inferior to the LT-88 as its theoretical coding gain drops quickly. At 0.25 bpp, on average, the LT-816 is about 1.8 dB worse than the LT-88. In the worst case, the LT-816 falls behind the LT-88 by nearly 4.5 dB in the highly textured *Barbara*. In our opinion, such large difference in *Barbara* is mainly due to 1) the design of the LT-816 is based on the simple AR(1) signal model. Although such a model works well for smooth images like *Lena*, it is not sufficient to describe highly textured image like *Barbara*. 2) As indicated in Fig. 9, the LT-816 has a very poor frequency resolution at high frequency bands, which cannot represent the rich texture components well.

C. Comparisons With the Downsampling/Interpolation and the Cascading Schemes

In the downsampling/interpolation and the cascading schemes, the size of the input image is also reduced before it is passed on to the DCT block coders. Thus, it is interesting to study the coding performance of different schemes when the complexities of the DCT coders are about the same (i.e., when the number of samples in the DCT codec is about the same). Specifically, we will compare the results of $k = 10/8 = 1.25$ and $k = 2$ with those of the LT-810 and the LT-816, respectively.

As can be seen from Table III and Figs. 11–13. The undersampled LT consistently outperforms the downsampling/interpolation scheme at various bit rates both in terms of the PSNR and the perceptual quality. The average PSNR is about 0.75 dB at 0.03125 bpp and more than 1.8 dB at 0.25 bpp. Furthermore, blocking artifacts are quite obvious in the downsampling/interpolation scheme, as shown in Fig. 12. This suggests that undersampled boundary pre-/postprocessing is more effective in reducing blocking artifacts. For the cascading scheme, the PSNR

Fig. 12. Reconstructed *Barbara* images coded at 0.03125 bpp ($\mathrm{CR} = 256 : 1$). First row (left): the original image; (middle): the JPEG2000 (21.88 dB); and (right): the LT-88 (20.99 dB). Second row (left): the downsampling/interpolation scheme with $k = 1.25$ (21.56 dB); (middle): the cascading scheme with $k = 1.25$ (22.18 dB);(right): the LT-810 (22.05 dB). Third row (left): the downsampling/interpolation scheme with $k = 2$ (21.90 dB); (middle): the cascading scheme with $k = 1.25$ (22.39 dB); (right): the LT-816 (22.43 dB).

results are comparable to those of the undersampled LTs at low bit rates. However, the performances of the cascading approach are worse with the increase of the bit rate. Fig. 11 evidently indicates that the undersampled LTs are applicable to a much wider range of bit rates at the same coding complexity. The visual quality of reconstructed images in Figs. 12 and 13 reveal similar facts. At very low bit rate, the perceptual performance for *Barbara* is about the same. At higher bit rates, it is obvious from Fig. 13 that the undersampled LTs are superior in preserving details such as the trousers region in *Barbara*.

Remarks: For the downsampling/interpolation and the cascading scheme, although improvements can be achieved by exploiting adaptive downsampling/interpolation filters [13], such systems incur high computational burden as the filters are signal dependent. Besides, we want to re-iterate here that the undersampled LTs are more suitable for hardware parallel processing as the pre-/postfiltering operations are conducted through the multiple input, multiple output matrix filter, rather than the single-input, single output scalar filter.

D. Selection of Pre-/Postfiltering Scheme in Practical Applications

Table III indicates that the best pre-/postfiltering scheme depends on both the input image and the bit rate. However, in practical low bit-rate compression, it is desirable to have a fixed pre-/postfiltering scheme exploitable for a wide-range of bit rates. Taking this fact into account, the LT-810 may be a good choice. Note that except for a few cases, the LT-810 provides the best or near optimal results for various images, as shown in Table III (the biggest loss is 0.84 dB for *Boat* coded at 0.03125 bpp). Besides, its encoding/decoding complexity is much lower than that of the critically sampled LT-88. Furthermore, it is also interesting to note that for all cases listed in Table III, the LT-810 combined with the L-CEB coder provides

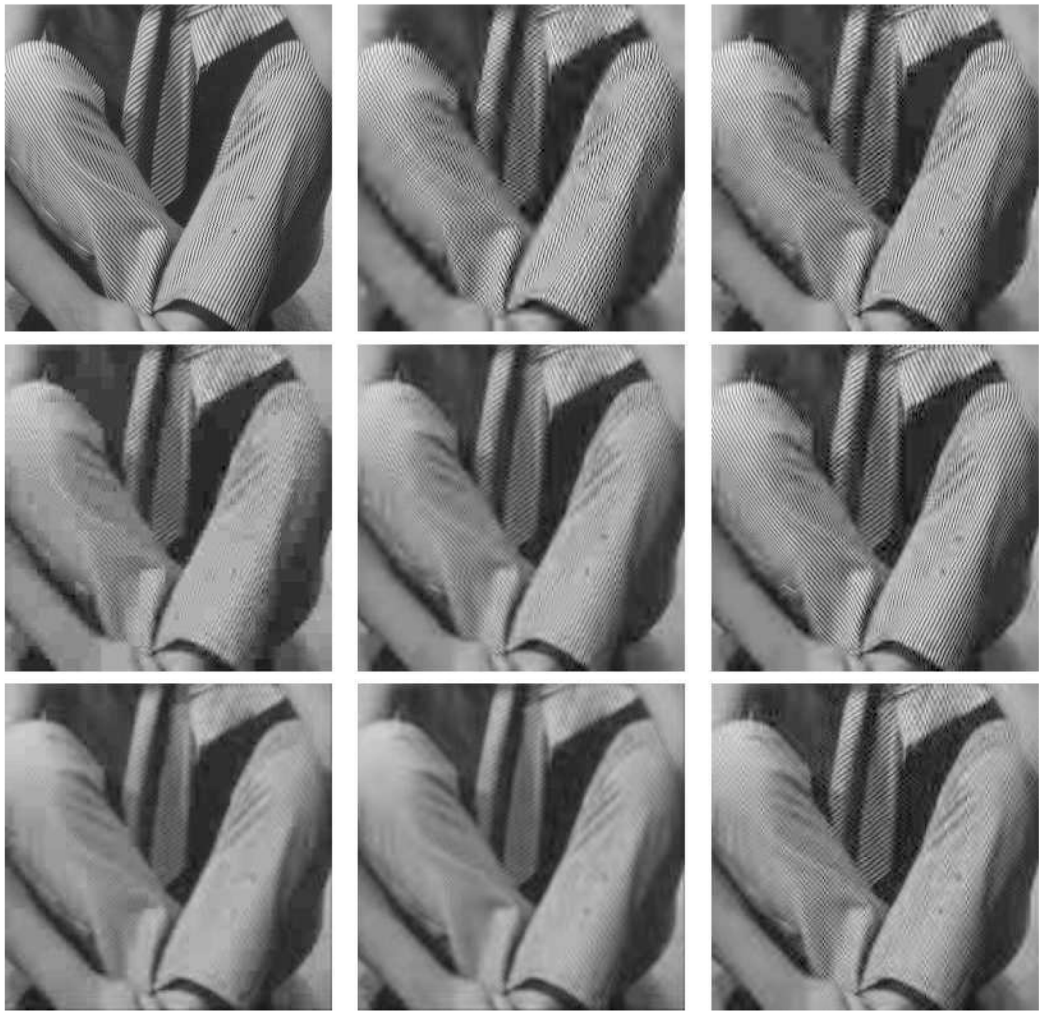


Fig. 13. Portions of the reconstructed *Barbara* images coded at 0.16 bpp ($\mathrm{CR} = 50{:}1$). First row (left): the original image; (middle): the JPEG2000 (26.25 dB); (right): the LT-88 (27.45 dB). Second row (left): the downsampling/interpolation scheme with $k = 1.25$ (25.09 dB); (middle): the cascading scheme with $k = 1.25$ (25.74 dB); (right): the LT-810 (27.54 dB). Third row (left): the downsampling/interpolation scheme with $k = 2$ (24.07 dB); (middle): the cascading scheme with $k = 1.2$ (24.20 dB); (right): the LT-816 (24.71 dB).

better PSNR results than the wavelet-based JPEG2000. In this light, the LT-810 proves itself as a high performance pre-/post-processing scheme with low coding/decoding complexity.

In some applications, e.g., sensor networks, the computing devices have low-power and small-memory requirements. The LT-816 may find applications in these devices as its codec only needs to process one-quarter amount of data as that of the original input. By using a decimator with $k = 2$, the conventional downsampling/interpolation and the cascading schemes can reduce the input data by the same amount. However, their performance drop quickly when the bit rate goes higher. Moreover, the undersampled boundary pre-/postfiltering scheme is based on nonoverlapped block operations, which is more suitable for hardware parallel implementation and buffer management.

VI. Conclusion

This paper presents the design and implementation of *undersampled* boundary per-/postfilters for block DCT coders, which are extensions of [8] to undersampled systems. Combined with the DCT/IDCT, the proposed systems can be viewed as undersampled lapped transforms or undersampled filter banks. Structures of undersampled pre-/postfilters are derived to yield the linear-phase property and they are pseudo inverse of each other with full rank. To further reduce the complexity in the optimization, pre-/postfilters are parameterized to minimize the reconstruction error when there is no quantization error. Two design examples are presented to verify the validity of the theoretical development. Image coding results are included to demonstrate the effectiveness of the proposed scheme over existing pre-/postfiltering schemes.

Appendix
Derivations of (19) and (20)

Before detailed derivations, we shall first introduce the generalized Rayleigh–Ritz theorem [23], which is crucial to the development.

Theorem 1: Let $\mathbf{X}$ be a $p \times p$ Hermitian matrix. For any $p \times r$ $(1 \leq r \leq p)$ orthonormal matrix $\mathbf{Y}$, we have

$$\min \mathrm{Trace}(\mathbf{Y}^T \mathbf{X} \mathbf{Y}) = \lambda_1(\mathbf{X}) + \lambda_2(\mathbf{X}) + \cdots + \lambda_r(\mathbf{X}) \tag{A1}$$

where $\lambda_i(\mathbf{X})$ (for $i = 1, \ldots, r$) is the ith smallest eigenvalue of $\mathbf{X}$. The above equality holds if the columns of $\mathbf{Y}$ are chosen to be orthonormal eigenvectors corresponding to r smallest eigenvalues of $\mathbf{X}$.

The derivation is based on the QR decomposition and the above theorem. First, using the QR decomposition [30], [31], the full rank matrices $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ can be always represented as

$$\hat{\mathbf{U}} = \hat{\mathbf{U}}_Q \begin{bmatrix} \hat{\mathbf{U}}_R \\ \mathbf{0} \end{bmatrix} \quad \text{and} \quad \hat{\mathbf{V}} = \hat{\mathbf{V}}_Q \begin{bmatrix} \hat{\mathbf{V}}_R \\ \mathbf{0} \end{bmatrix} \tag{A2}$$

where $\hat{\mathbf{U}}_Q$ and $\hat{\mathbf{V}}_Q$ are $m \times m$ orthonormal matrices, while $\hat{\mathbf{U}}_R$ and $\hat{\mathbf{V}}_R$ are $n \times n$ upper triangular matrices with nonzero diagonal entries. Using (A2), $\mathbf{U}$ and $\mathbf{V}$ in (13) can be expressed as

$$\mathbf{U} = [\,\hat{\mathbf{U}}_R^{-1} \quad \mathbf{0}\,]\,\hat{\mathbf{U}}_Q^T \quad \text{and} \quad \mathbf{V} = [\,\hat{\mathbf{V}}_R^{-1} \quad \mathbf{0}\,]\,\hat{\mathbf{V}}_Q^T. \tag{A3}$$

Let $\hat{\mathbf{U}}_Q$ and $\hat{\mathbf{V}}_Q$ be further partitioned into $\hat{\mathbf{U}}_Q = [\hat{\mathbf{U}}_{Q,l} \;\; \hat{\mathbf{U}}_{Q,r}]$ and $\hat{\mathbf{V}}_Q = [\hat{\mathbf{V}}_{Q,l} \;\; \hat{\mathbf{V}}_{Q,r}]$, where $\hat{\mathbf{U}}_{Q,l}$ and $\hat{\mathbf{V}}_{Q,l}$ are of size $m \times n$ each, and $\hat{\mathbf{U}}_{Q,r}$ and $\hat{\mathbf{V}}_{Q,r}$ are of size $m \times (n-m)$ each. Based on such partitions, it can be readily seen that (A2)–(A3) can be simplified into

$$\hat{\mathbf{U}} = \hat{\mathbf{U}}_{Q,l}\hat{\mathbf{U}}_R, \quad \hat{\mathbf{V}} = \hat{\mathbf{V}}_{Q,l}\hat{\mathbf{V}}_R \tag{A4}$$

$$\mathbf{U} = \hat{\mathbf{U}}_R^{-1}\hat{\mathbf{U}}_{Q,l}^T, \quad \mathbf{V} = \hat{\mathbf{V}}_R^{-1}\hat{\mathbf{V}}_{Q,l}^T. \tag{A5}$$

Then, substituting (A4) and (A5) into $\mathbf{T}$ and $\mathbf{P}$, respectively, and using the equalities $\hat{\mathbf{U}}_{Q,l}^T\hat{\mathbf{U}}_{Q,l} + \hat{\mathbf{U}}_{Q,r}^T\hat{\mathbf{U}}_{Q,r} = \mathbf{I}$ and $\hat{\mathbf{V}}_{Q,l}^T\hat{\mathbf{V}}_{Q,l} + \hat{\mathbf{V}}_{Q,r}^T\hat{\mathbf{V}}_{Q,r} = \mathbf{I}$, we arrive at the following expression of $\mathbf{E} = \mathbf{I} - \mathbf{TP}$:

$$\mathbf{E} = \mathbf{W}_M \begin{bmatrix} \hat{\mathbf{U}}_{Q,r}\hat{\mathbf{U}}_{Q,r}^T & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{V}}_{Q,r}\hat{\mathbf{V}}_{Q,r}^T \end{bmatrix} \mathbf{W}_M. \tag{A6}$$

By (15) and (A6), σ_r^2 can be re-written into

$$\sigma_r^2 = \mathrm{Trace}\left(\mathbf{W}_M \begin{bmatrix} \hat{\mathbf{U}}_{Q,r}\hat{\mathbf{U}}_{Q,r}^T & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{V}}_{Q,r}\hat{\mathbf{V}}_{Q,r}^T \end{bmatrix} \times \mathcal{R}'_{xx} \begin{bmatrix} \hat{\mathbf{U}}_{Q,r}\hat{\mathbf{U}}_{Q,r}^T & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{V}}_{Q,r}\hat{\mathbf{V}}_{Q,r}^T \end{bmatrix} \mathbf{W}_M \right) \tag{A7}$$

where $\mathcal{R}'_{xx} = \mathbf{W}_M\mathcal{R}_{xx}\mathbf{W}_M$. Using the fact that $\mathrm{Trace}(\mathbf{XY}) = \mathrm{Trace}(\mathbf{YX})$, along with the orthonormal properties of $\mathbf{W}_M, \hat{\mathbf{U}}_{Q,r}$, and $\hat{\mathbf{V}}_{Q,r}$, one can reduce σ_r^2 to

$$\begin{aligned} \sigma_r^2 &= \mathrm{Trace}\left(\begin{bmatrix} \hat{\mathbf{U}}_{Q,r}^T & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{V}}_{Q,r}^T \end{bmatrix} \mathcal{R}'_{xx} \begin{bmatrix} \hat{\mathbf{U}}_{Q,r} & \mathbf{0} \\ \mathbf{0} & \hat{\mathbf{V}}_{Q,r} \end{bmatrix} \right) \\ &= \mathrm{Trace}\left(\hat{\mathbf{U}}_{Q,r}^T\mathcal{R}_u\hat{\mathbf{U}}_{Q,r} \right) + \mathrm{Trace}\left(\hat{\mathbf{V}}_{Q,r}^T\mathcal{R}_v\hat{\mathbf{V}}_{Q,r} \right) \end{aligned} \tag{A8}$$

where $\mathcal{R}_u$ and $\mathcal{R}_v$ are the $m \times m$ upper-left and lower-right submatrices of $\mathcal{R}'_{xx}$, as defined in (16).

Next, by applying Theorem 1 to (A8), we know that

$$\min \mathrm{Trace}\left(\hat{\mathbf{U}}_{Q,r}^T\mathcal{R}_u\hat{\mathbf{U}}_Q \right) = \sum_{i=1}^{m-n} \lambda_i(\mathcal{R}_u) \tag{A9}$$

and

$$\min \mathrm{Trace}\left(\hat{\mathbf{V}}_{Q,r}^T\mathcal{R}_v\hat{\mathbf{V}}_Q \right) = \sum_{i=1}^{m-n} \lambda_i(\mathcal{R}_v). \tag{A10}$$

Therefore, the minimal value of σ_r^2 is as in (19).

Now, what remains is the characterizations of $\hat{\mathbf{U}}$ and $\hat{\mathbf{V}}$ so that (19) can be achieved. Note that equalities (A9) and (A10) hold when the columns of $\hat{\mathbf{U}}_{Q,r}$ and $\hat{\mathbf{V}}_{Q,r}$ are the orthonormal eigenvectors corresponding to $m-n$ smallest eigenvalues of $\mathcal{R}_u$ and $\mathcal{R}_v$, respectively. From (18), it is obvious that $\mathbf{A}_u(:, i)$ is the orthonormal eigenvector of $\mathcal{R}_u$ associated with $\lambda_i(\mathcal{R}_u)$. Similar arguments can be applied to $\mathbf{A}_v$ and $\mathcal{R}_v$. Hence, to get (A10), $\hat{\mathbf{U}}_{Q,r}$ and $\hat{\mathbf{V}}_{Q,r}$ can be chosen as

$$\begin{aligned} \hat{\mathbf{U}}_{Q,r} &= \mathbf{A}_u(:, 1 : m-n) \\ \hat{\mathbf{V}}_{Q,r} &= \mathbf{A}_v(:, 1 : m-n). \end{aligned} \tag{A11}$$

Recall that in (A4), the components $\hat{\mathbf{U}}_{Q,l}$ and $\hat{\mathbf{V}}_{Q,l}$ are orthogonal complements of $\hat{\mathbf{U}}_{Q,r}$ and $\hat{\mathbf{V}}_{Q,r}$, respectively. When $\hat{\mathbf{U}}_{Q,r}$ and $\hat{\mathbf{V}}_{Q,r}$ are chosen as in (A11), it is obvious that the columns of $\mathbf{A}_u(:, m-n+1 : m)$ and $\mathbf{A}_v(:, m-n+1 : m)$ are orthonormal bases for the complements of $\hat{\mathbf{U}}_{Q,r}$ and $\hat{\mathbf{V}}_{Q,r}$, respectively. Hence, $\hat{\mathbf{U}}_{Q,l}$ and $\hat{\mathbf{V}}_{Q,l}$ can be completely expressed as [23]

$$\begin{aligned} \hat{\mathbf{U}}_{Q,l} &= \mathbf{A}_u(:, m-n+1 : m)\hat{\mathbf{U}}_q \\ \hat{\mathbf{V}}_{Q,l} &= \mathbf{A}_v(:, m-n+1 : m)\hat{\mathbf{V}}_q \end{aligned} \tag{A12}$$

where $\hat{\mathbf{U}}_q$ and $\hat{\mathbf{V}}_q$ are arbitrary $n \times n$ orthonormal matrices. Finally, substituting (A12) into (A4) yields (20).

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their thorough reviews and numerous constructive suggestions which significantly enhance the presentation of the paper.

REFERENCES

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

[2] C. J. Kuo and R. J. Hsieh, "Adaptive postprocessor for block encoded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, no. 11, pp. 1271–1283, Nov. 1995.

[3] J. Chou, M. Crouse, and K. Ramchandran, "A simple algorithm for removing blocking artifacts in block-transform coded images," *IEEE Signal Process. Lett.*, vol. 5, no. 2, pp. 33–35, Feb. 1998.

[4] D. Yula and H. Webb, "Image restoration by the method of convex projections: Part I-theory," *IEEE Signal Process. Lett.*, vol. 5, no. 2, pp. 33–35, Feb. 1998.

[5] A. Zakhor, "Iterative procedures for reduction of blocking effects in transform image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 2, no. 3, pp. 91–95, Mar. 1992.

[6] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, "Regularizd reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 12, pp. 421–432, Dec. 1993.

[7] T. D. Tran, "Lapped transform via time-domain pre- and post-processing," in *Proc. 5th Annu. Conf. Information Sciences and Systems*, Baltimore, MD, Mar. 2001, pp. 890–895.

[8] T. D. Tran, J. Liang, and C. Tu, "Lapped transform via time-domain pre- and post-filtering," *IEEE Trans. Signal Process.*, vol. 51, no. 6, pp. 1557–1571, Jun, 2003.

[9] B. Zeng and A. Venetsanopoulos, "A JPEG-based interplotive image coding scheme," in *Proc. IEEE ICASSP*, Minnneapolis, MN, Apr. 1993, pp. 393–396.

[10] G. Aharoni, A. Averbuch, R. Coifman, and M. Israeli, "Local cosine transform a method for the reduction of the blocking effect in JPEG," *J. Math. Imag. Vis.*, vol. 3, pp. 7–38, 1993.
[11] A. Averbuch, A. Schclar, and D. Donoho, "Deblocking of block-transform compressed images using weighted sums of symmetrically aligned pixels," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 200–212, Feb. 2005.
[12] A. M. Bruckstein, M. Elad, and R. Kimmel, "Down-scaling for better transform compression," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1132–1144, Sep. 2003.
[13] Y. Tsaig, M. Elad, and P. Milanfar, "Variable projection for near-optimal filtering in low bit-rate block coders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 154–160, Jan. 2005.
[14] C. Tu and T. D. Tran, "Context based entropy encoding of block coefficient transforms for image compression," *IEEE Trans. Image Process.*, vol. 11, no. 11, pp. 1271–1283, Nov. 2002.
[15] ——, L-CEB Coder [Online]. Available: http://thanglong.ece.jhu.edu/cjtu/project.html 2002
[16] D. L. M. J. Gonnish and M. W. Lee, "JPEG 2000: Overview, architecture, and applications," in *Proc. IEEE Int. Conf. Image Processing*, Vancouver, BC, Canada, Oct. 2000, pp. 29–32.
[17] C. C. A. Skodras and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Process. Mag.*, vol. 5, no. 9, pp. 36–58, Sep. 2001.
[18] H. S. Malvar and D. H. Staelin, "The LOT: Transform coding without blocking effects," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 2, pp. 553–559, Apr. 1989.
[19] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 3, pp. 969–978, Jun. 1990.
[20] ——, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Trans. Signal Process.*, vol. 46, no. 4, pp. 1043–1053, Apr. 1998.
[21] C. Tu, T. D. Tran, and J. Liang, "Over-sampled and under-sampled pre-post-filters for block dct coders," in *Proc. IEEE Int. Conf. Image Processing*, Singapore, Oct. 2004, pp. 1277–1280.
[22] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[23] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York: Cambridge Univ. Press, 1991.
[24] L. Peter and M. Timenetsky, *The Theory of Matrices with Applications*. Orlando, FL: Academic, 1985.
[25] T. D. Tran, "M-channel linear phase perfect reconstruction filter bank with rational coefficients," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 49, no. 7, pp. 914–927, Jul. 2002.
[26] T. A. Ramstad, S. O. Aase, and J. H. Husoy, *Subband Compression of Images: Principles and Examples*. New York: Elsevier, 1995.
[27] S. O. Aase and T. Ramstad, "On the optimality of nonunitary filter banks in subband coders," *IEEE Trans. Image Process.*, vol. 4, no. 12, pp. 1585–1591, Dec. 1995.
[28] J. Katto and Y. Yasuda, "Performance evaluation of subband coding and optimization of its filter coefficients," *SPIE Proc. Vis. Commun. Image Process.*, pp. 95–106, Nov. 1991.
[29] D. Taubman, Kakadu Software 2006 [Online]. Available: http://www.kakadusoftware.com/
[30] E. Chu and A. George, "QR factorization of a dense matrix on a hypercube multiprocessor," *SIAM J. Sci. Statist. Comput.*, vol. 11, pp. 990–1028, 1990.
[31] A. Pothen and P. Raghavan, "Distributed orthogonal factorization: Givens and householder algorithms," *SIAM J. Sci. Statist. Comput.*, vol. 10, pp. 1113–1134, 1989.

Lu Gan (S'02–M'04) photograph and biography not available at the time of publication.

Chengjie Tu (S'02–M'04) photograph and biography not available at the time of publication.

Jie Liang (S'99–M'04) photograph and biography not available at the time of publication.

Trac D. Tran (S'94–M'98) photograph and biography not available at the time of publication.

Kai-Kuang Ma (S'80–M'84–SM'95) photograph and biography not available at the time of publication.