# Lapped Transform via Time Domain Pre- and Post-Filtering. Part II: Fast Algorithms

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

*Abstract*— **In a companion paper [1], we introduce a general framework for the design and implementation of a large family of lapped transforms via time-domain pre-filtering of the DCT input and post-filtering of the IDCT output. In this follow-up paper, we first present the optimized orthogonal and biorthogonal results for the time domain pre- and post-filtering lapped transform (TDLT) obtained from unconstrained optimization. Several simplified models are then developed to approximate the optimal pre- and post-filter in the TDLT. These fast and efficient models are based on cascades of plane rotation operators and lifting steps, respectively. Despite tremendous savings in computational complexity, the optimized results corresponding to these simplified pre/post-filters are virtually identical to that of the complete solution. The multiplierless versions of these pre- and post-filters when combined with an appropriate multiplierless block transform such as the binDCT [2] generate a family of VLSI-friendly fast lapped transforms with reversible integer-to-integer mapping. These properties make the proposed fast TDLT suitable for many practical applications.**

*Keywords*— **Lapped transform, pre- and post-filtering, DCT, lifting scheme, fast and multiplierless algorithms.**

## I. INTRODUCTION

THE companion Part I paper presents a general framework of block-based signal decomposition and representation via a modular cascade of block operators [1]. A particular solution in this framework can be viewed as pre- and post-filtering along the block boundaries of a traditional block-based DCT decomposition scheme. The pre- and post-filters operate in time domain and outside the existing block-based infrastructure as illustrated in Fig. 1(a). The corresponding polyphase representation is depicted in Fig. 1(c). This framework can also be viewed as a particular implementation of an $M$-band filter bank or lapped transform.

The pre-filtering operator $\mathbf{P}$ processes the block boundaries, removing inter-block correlation. The pre-processed samples are then fed to the DCT to be transformed and encoded as usual. At the decoder side, the corresponding inverse operator $\mathbf{P}^{-1}$ serves as the post-filter, reconstructing the data in an overlapping manner, minimizing or eliminating blocking artifacts. Our philosophy is that as long as there is a smooth and continuous transition from block to block, blocking artifacts will not be visible.

Moreover, we have also shown that there is an intimate connection between the popular type-II fast lapped orthogonal transform (LOT-II) [3] and the time-domain pre-/post-filtering of block DCT's input samples and block IDCT's output samples [1]. The pre-filtering block operator derived from the LOT-II is shown in Fig. 1(c). It consists of two stages of butterflies and a matrix $\mathbf{V}$ between them:

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

where $\mathbf{I}$, $\mathbf{J}$, and $\mathbf{0}$ are the $\frac{M}{2} \times \frac{M}{2}$ identity matrix, the reversal matrix, and the null matrix, respectively ($M$ is the block size). The matrix $\mathbf{V}$, controlling pre- and post-filtering, is given by

$$\mathbf{V} = \mathbf{J}\,\mathbf{C}_{M/2}^{II^T}\,\mathbf{S}\,\mathbf{C}_{M/2}^{IV}\,\mathbf{J}, \quad (2)$$

where $\mathbf{C}_{M/2}^{II}$ is the $\frac{M}{2}$-point type-II DCT matrix, $\mathbf{C}_{M/2}^{IV}$ is the $\frac{M}{2}$-point type-IV DCT matrix, and $\mathbf{S}$ is a diagonal matrix that introduces biorthogonality. One example of $\mathbf{S}$ is $diag\{s, 1, \ldots, 1\}$; the scaling factor $s$ is chosen as the Golden Ratio $\frac{1+\sqrt{5}}{2}$ or its approximation. The structure produces orthogonal basis functions if $\mathbf{S}$ is chosen to be the identity matrix ($s = 1$).

To maintain perfect reconstruction, the post-filtering operator has to be the exact inverse of the pre-filtering operator. Hence, the corresponding post-filtering operator is

$$\mathbf{P}^{-1} = \frac{1}{2}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{V}^{-1} \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{J} \\ \mathbf{J} & -\mathbf{I} \end{bmatrix}. \quad (3)$$

Its complexity is the same as the pre-filter's.

The proposed pre/post-processing framework is quite flexible. Complexity versus performance trade-off can be achieved by simply varying the size of $\mathbf{P}$, i.e., varying the number of borrowed samples at block boundaries. In other words, the following choice of $\mathbf{V}$ involves $N$ ($N \leq \frac{M}{2}$) samples borrowed at each block boundary

$$\mathbf{V} = \begin{bmatrix} \mathbf{J}\,\mathbf{C}_N^{II^T}\,\mathbf{S}\,\mathbf{C}_N^{IV}\,\mathbf{J} & \mathbf{0}_{N \times \frac{M-2N}{2}} \\ \mathbf{0}_{\frac{M-2N}{2} \times N} & \mathbf{I}_{\frac{M-2N}{2}} \end{bmatrix}. \quad (4)$$

Since the lapped transform described in (1)-(4) is obtained from time-domain pre- and post-filtering, we label it the type-I time-domain lapped transform (TDLT-I). The type-I classification distinguishes it from other designs that we shall present later in the paper. The TDLT-I provides several advantages: *(i)* it has an elegant closed-form expression; *(ii)* it improves the coding efficiency of the DCT as well as eliminates the annoying blocking artifacts by applying extra pre- and post-filtering across boundaries of neighboring blocks; and *(iii)* its time-domain nature enables the TDLT-I to be implemented on the basis of the existing DCT-based image and video coding framework with minimal software/hardware upgrade.

The authors are with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218. Email: {jieliang, cjtu, trac}@jhu.edu.

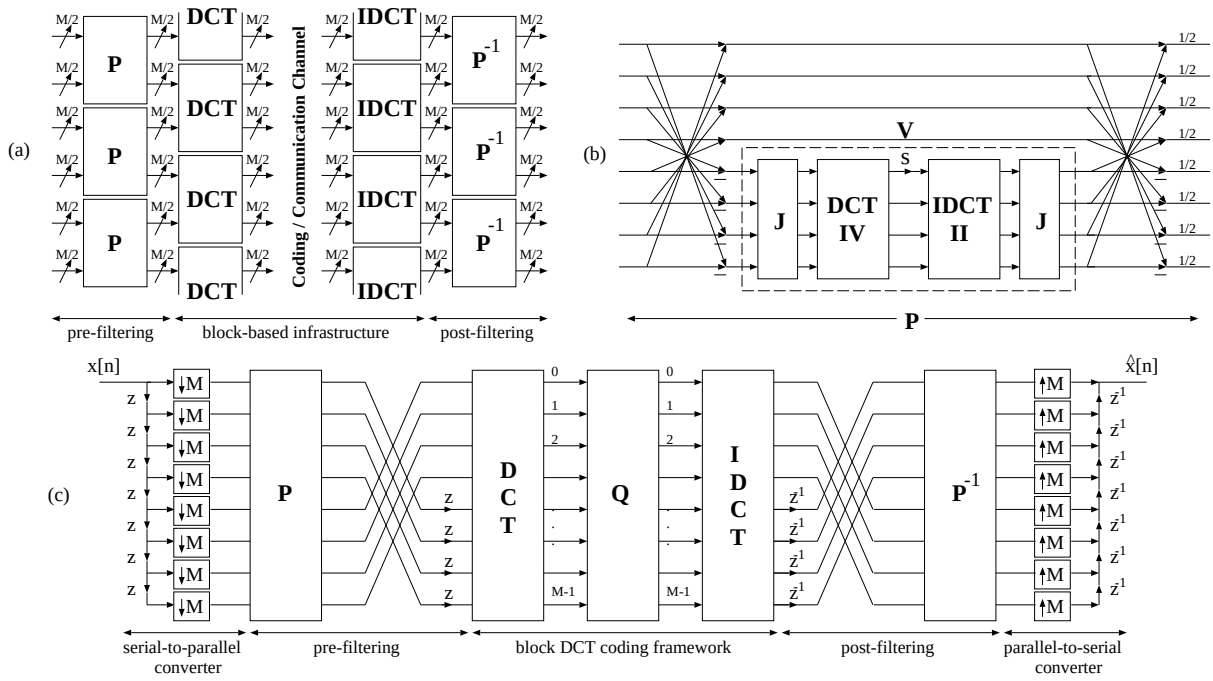


Fig. 1. General structure of the TDLT. (a) Global view. (b) The pre-filter $\mathbf{P}$ derived from the LOT-II. (c) Polyphase structure.

However, despite employing components with fast-computable algorithms such as the type-II and the type-IV DCT, the pre- and post-filter in the TDLT-I are still computationally expensive. Obviously, this is undesirable in many critical real-time applications. The complexity analysis in [1] shows that the computational overhead is slightly above 100% in the full-borrowing case ($N = \frac{M}{2}$). When the number of borrowed samples is small ($N \leq 4$), there is also inefficiency in implementing $\mathbf{V}$ as a cascade of DCT-IV and IDCT-II rather than direct matrix multiplication. Lastly, floating-point multiplications are still required in the TDLT-I, which are costly in both hardware and software implementations.

In this paper, we will first present the optimized orthogonal and biorthogonal pre/post-filter in the coding-gain sense that can be obtained within the TDLT framework. This result confirms that the TDLT-I presented in the first part of this paper is indeed near optimal. Next, observing the behaviors of the optimized results as well as the TDLT-I leads us to several simplified and robust models of the pre/post-filter in the TDLT. These fast efficient designs are based on cascading a minimal number of either rotation angles or lifting steps [4], [5] in similar fashion as the type-I fast LOT (LOT-I) [6], [3] or the LiftLT [7]. However, better performance and flexibility can be achieved in the new framework.

Finally, the proposed low-complexity pre/post-filter can be combined with existing fast algorithms for the DCT [8], [9], [10], [11], [12], [13] to achieve fast and high-performance signal decomposition. In particular, the binDCT, a lifting-based multiplierless approximation of the DCT, will be employed in this paper [14], [2]. Multiplierless transform and lossless transform can be easily achieved with the binDCT and our lifting-based pre/post-filters.

The paper is organized as follows. In Section II, we present the optimized TDLT results obtained from unconstrained optimization. The parameterization of arbitrary orthogonal and invertible matrix will also be reviewed. Section III introduces the plane rotation-based fast orthogonal TDLT, followed by a lifting-based simplified model in Section IV. Some alternative structures for the lifting-based pre/post-filter are presented in Section V. All aforementioned pre/post-filters can be employed at block boundaries of the DCT of odd block sizes as well. The resulting odd-channel TDLTs are demonstrated in Section VI. Next, we summarize the computational complexity of various fast TDLTs in Section VII and present some rational as well as dyadic approximations of various optimized TDLT parameters in Section VIII. Finally, a practical image coding experiment is presented in Section IX, followed by the conclusion in Section X. We use the same notations as in [1].

## II. The Optimized TDLT

In this section, we present the highest coding gains that can be achieved when the matrix $\mathbf{V}$ in the TDLT is allowed to be any orthogonal or invertible matrix. The solution is obtained from unconstrained optimization where the methods chosen to parameterize the desired orthogonal or invertible matrix $\mathbf{V}$ are described in the following section.

### A. Matrix Parameterization

It is well known that any $N \times N$ orthogonal matrix can be factored as a cascade of $N(N-1)/2$ plane rotations and $N$ sign parameters [15]. The factorization procedure is summarized as follows. A plane rotation operator $\mathbf{R}_{ij}(\theta_k)$,

$j > i$, is defined as an $N \times N$ identity matrix with its $(i,i)$, $(i,j)$, $(j,i)$, and $(j,j)$-th entries replaced by:

$$\begin{bmatrix} cos(\theta_k) & sin(\theta_k) \\ -sin(\theta_k) & cos(\theta_k) \end{bmatrix} \tag{5}$$

It can be easily verified that the plane rotation operator is orthogonal, i.e., $\mathbf{R}_{ij}^{-1}(\theta_k) = \mathbf{R}_{ij}^{T}(\theta_k)$.

An orthogonal matrix can be turned into a diagonal matrix after sequentially pre- or post-multiplying the matrix with suitable rotation operators. In the following analysis, we will pre-multiply the given matrix and eliminate its off-diagonal entries column-wise from the left to the right. In this case, the rotation angle $\theta_k$ in an operator $\mathbf{R}_{ij}(\theta_k)$ is selected such that the $(j,i)$-th entry of the resulting matrix is eliminated.

Let $\mathbf{V}$ be an $N \times N$ orthogonal matrix, and define $\mathbf{R}_{ij}(\theta_k)\,\mathbf{V} = \mathbf{B}$. The $(j,i)$-th entry of $\mathbf{B}$ can be written as:

$$\mathbf{B}_{j,i} = -sin(\theta_k)\,\mathbf{V}_{i,i} + cos(\theta_k)\,\mathbf{V}_{j,i}. \tag{6}$$

Therefore in order to get $\mathbf{B}_{j,i} = 0$, the angle $\theta_k$ should be chosen such that

$$tan(\theta_k) = \frac{\mathbf{V}_{j,i}}{\mathbf{V}_{i,i}},\ \theta_k \in [-\pi/2, \pi/2]. \tag{7}$$

By this selection of the rotation angles, after pre-multiplying the matrix $\mathbf{V}$ by $\mathbf{R}_{0,1}(\theta_0)$, $\mathbf{R}_{0,2}(\theta_1)$, ... $\mathbf{R}_{0,N-1}(\theta_{N-2})$ (row and column indices start from zero), we obtain a new orthogonal matrix

$$\mathbf{C} = \mathbf{R}_{0,N-1}(\theta_{N-2}) \ldots \mathbf{R}_{0,2}(\theta_1)\,\mathbf{R}_{0,1}(\theta_0)\,\mathbf{V}. \tag{8}$$

All off-diagonal entries in the first column of $\mathbf{C}$ become zero. Thus, the diagonal entry $\mathbf{C}_{0,0}$ must be 1 or $-1$, since $\mathbf{C}$ is an orthogonal matrix. Similarly, all off-diagonal entries in the first row of $\mathbf{C}$ would also be zero.

The same approach can be applied to the remaining columns of the resulting matrix. As a summary, let $\mathbf{R}_i$ represents the $i$-th plane rotation operators, the whole factorization can be written as:

$$\mathbf{R}_{L-1} \ldots \mathbf{R}_1\,\mathbf{R}_0\,\mathbf{V} = \Lambda, \tag{9}$$

where $L = N(N-1)/2$ and $\Lambda$ is a diagonal matrix with either 1 or $-1$ as its diagonal entries. From this, we can represent $\mathbf{V}$ as

$$\mathbf{V} = \mathbf{R}_0{}^T\,\mathbf{R}_1{}^T \ldots \mathbf{R}_{L-1}{}^T\,\Lambda. \tag{10}$$

The flow-graph of this representation for a $4 \times 4$ matrix $\mathbf{V}$ is shown in Fig. 2(a). Note that this representation is highly non-unique, since we can choose between the pre- or post-multiplying approach, and in each case, either forward or backward elimination can be used. As an example, $\mathbf{V}$ can also be represented as in Fig. 2(b) if the rotations in the pre-multiplication approach are designed such that the column-wise elimination starts from right to left.

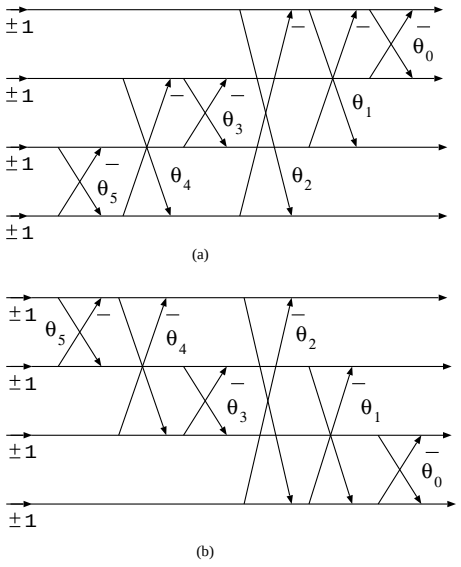


Fig. 2. Representation of an orthogonal matrix by rotation angles: (a) with pre-multiplying and forward elimination; (b) with pre-multiplying and backward elimination.

An optimization program can use (10) to parameterize an arbitrary orthogonal matrix. To model an invertible matrix, we can use the method of singular value decomposition (SVD) [16], which shows that an invertible matrix can be represented by $\mathbf{USW}$, where $\mathbf{U}$ and $\mathbf{W}$ are orthogonal matrices and $\mathbf{S}$ is a diagonal matrix with nonzero diagonal entries. The orthogonal matrices $\mathbf{U}$ and $\mathbf{W}$ can be parameterized by rotation angles, whereas the diagonal matrix $\mathbf{S}$ can be represented by its nonzero scaling factors.

### *B. Optimization*

Based on these modelings, an unconstrained optimization program is set up to find the optimal coding gain of the orthogonal or biorthogonal $M \times (M + 2N)$ TDLT, where $N \leq \lfloor M/2 \rfloor$. The selected optimization routine is a simplex-based method, which cannot guarantee the global optimality of the result. Therefore, initial values play an important role in the optimization, especially for large $N$. The initial values of the rotation angles and sign parameters in the model are obtained from the matrix $\mathbf{V}$ in the TDLT-I. In fact, the sign parameters can be fixed as that of the TDLT-I without affecting the final solution, since the performance of the TDLT-I is quite close to the optimal. When the sign parameters are ignored, the free parameters for the orthogonal pre-filter are the $N(N-1)/2$ rotation angles. In the biorthogonal case, there are $N(N-1)$ rotation angles and $N$ diagonal entries.

Table I compares the coding gains of the optimized orthogonal and biorthogonal TDLT (denoted as TDLOT and TDLT, respectively), the TDLT-I and the optimal LOT, where the results of TDLT-I are obtained by (2) with a scaling factor of 8/5, and the optimal LOT are obtained by replacing the matrix $\mathbf{V}$ in the LOT structure with the corresponding Karhunen-Loève transform (KLT) [3].

It can be seen that except for the case of $M = 4$, the optimized TDLOT achieves higher coding gain than the

TABLE I
CODING GAIN IN DB OF VARIOUS LAPPED TRANSFORM FOR AN AR(1) SIGNAL WITH $\rho = 0.95$.

| Size | TDLOT-I & LOT-II | Opt. LOT | Opt. TDLOT | TDLT-I | Opt. TDLT |
|---|---|---|---|---|---|
| $4 \times 6$ | 7.5701 | - | 7.5701 | 8.0683 | 8.0685 |
| $4 \times 8$ | 7.9259 | 7.9479 | 7.9426 | 8.5652 | 8.6349 |
| $8 \times 10$ | 8.8259 | - | 8.8259 | 9.0600 | 9.0611 |
| $8 \times 12$ | 8.9910 | - | 9.0015 | 9.3051 | 9.3375 |
| $8 \times 14$ | 9.1122 | - | 9.1397 | 9.4489 | 9.5033 |
| $8 \times 16$ | 9.2189 | 9.2357 | 9.2592 | 9.5492 | 9.6151 |
| $16 \times 32$ | 9.7593 | 9.7702 | 9.8079 | 9.9079 | 9.9597 |
| $32 \times 64$ | 9.9729 | 9.9815 | 10.0145 | 10.0279 | 10.0667 |

optimal LOT. Its coding gain is $9.2592\,dB$ in the $8 \times 16$ case, while the optimal LOT achieves only $9.2357\,dB$. The optimized biorthogonal TDLT has much higher coding gain than the orthogonal case. For instance, the $8 \times 16$ optimized TDLT has a coding gain of $9.6151\,dB$, which is impressively close to the optimal $9.63\,dB$ in [17] and the optimized GLBT in [18]. However, the TDLT has a much simpler structure than both of the above.

Table I also shows that the coding gain of the TDLT-I, which is identical to that of the LOT-II, is only below the optimized case by about $0.04\,dB$, implying that it is a very robust approximation of the optimal results. Also, note that the optimal $M \times (M+2)$ orthogonal TDLT with linear phase actually reduces to the $M \times M$ DCT, since in this case, $\mathbf{V}$ is a scalar, which can only be 1 or $-1$ (the former choice always gives higher coding gain than the latter). The corresponding $2 \times 2$ pre-filter thus reduces to the identity matrix.

The frequency responses of some optimized TDLTs are shown in Fig. 3 to Fig. 5 together with their impulse responses. The synthesis filters of the biorthogonal TDLT are smoother than the analysis filters – the key to blocking artifact elimination in low bit-rate coding applications.

## III. Plane Rotation-based Fast TDLOT-II

It is observed that the significant entries of the matrix $\mathbf{V}$ in the optimized pre-filter concentrate along the three main diagonals. This is also true for the matrix $\mathbf{V}$ in the TDLT-I, given by the inverse DCT-II and DCT-IV. Therefore it can be expected that many parameters in the rotation angle-based representation are very small. For example, the $4 \times 4$ matrix $\mathbf{V}$ in TDLOT-I is

$$\mathbf{V} = \begin{bmatrix} 0.8072 & 0.5594 & 0.1436 & 0.1218 \\ -0.5718 & 0.6992 & 0.4214 & 0.0814 \\ 0.1218 & -0.4443 & 0.8600 & 0.2193 \\ -0.0814 & -0.0286 & -0.2492 & 0.9646 \end{bmatrix}.$$

With forward elimination, the rotation angles $\theta_0$ to $\theta_5$ in Fig. 2(a) for this matrix are found to be

$$\theta = [-0.20\pi,\ 0.04\pi,\ -0.03\pi,\ -0.15\pi, -0.01\pi,\ -0.08\pi].$$

Since $\theta_1$, $\theta_2$ and $\theta_4$ are relatively small, we expect that they can be discarded without significant performance loss. Another observation is that the remaining angles have a strong decreasing trend: the magnitudes of the off-diagonal entries in $\mathbf{V}$ decrease steadily from the upper left corner to the lower right corner.

The strong tridiagonal property of the optimized matrix $\mathbf{V}$ suggests a simplified orthogonal model for the matrix $\mathbf{V}$: a cascade of rotation angles between neighboring channels. The corresponding TDLT structure is shown in Fig. 6 for the case of $M = 8$. We denote this model as the TDLOT-II. For an $M \times (M + 2N)$ TDLOT, this simplified model only needs $N - 1$ rotation angles. Comparing to the TDLOT-I, the complexity of this algorithm is reduced significantly, enabling much faster implementation.

If backward elimination is used, we arrive at a similar model for $\mathbf{V}$ after removing trivial angles in Fig. 2(b). This is exactly the same structure used in the type-I fast LOT [3], [6]. The difference between the two cases is that in the LOT-I, the cascading of rotation angles starts from the top channels and propagates to the bottom of $\mathbf{V}$, while in the TDLT-II depicted in Fig. 6, it starts from the bottom and propagates to the top. The significance of this difference will be explained later in this section.

Coding gain results of the TDLOT-II are presented in Table II, which also contains results of other fast TDLTs presented in the remaining sections. The results of the initial TDLOT-II are obtained by factoring the TDLOT-I with forward elimination and then keeping only the non-trivial angles. The optimized TDLOT-II are obtained by further optimizing the remaining rotation angles in Fig. 6.

By comparing Table I and Table II, it is clear that the performance of the optimized TDLOT-II is very close to that of the optimized TDLOT with full matrix model. For example, the coding gains of the optimized $8 \times 16$ and $16 \times 32$ TDLOT are 9.2592 and $9.8079\,dB$, while the optimized TDLOT-II can achieve 9.2563 and $9.8014\,dB$, respectively. In fact, the initial TDLOT-II results obtained directly from the TDLOT-I are already very close to the optimized cases. More interestingly, they yield even higher coding gains then their TDLOT-I originals. All of these show that the type-II simplified model is a very accurate approximation of the optimal results.

Table III lists the rotation angles in several optimized TDLOT-II. As previously mentioned, their magnitudes are steadily decreasing. The reason is quite intuitive. The pre-filter in the TDLT framework is applied at the boundaries of neighboring signal blocks, and it tries to smoothen the input to the DCT in order to improve the energy compaction. From Fig. 1 and Fig. 6, because of the first stage of butterflies, each input to the matrix $\mathbf{V}$ is the difference between a pair of samples from two sides of a block boundary. Moreover, the upper inputs of $\mathbf{V}$ correspond to the differences of nearer samples, whereas the lower inputs correspond to those of farther samples. Since the correlations between nearer neighbors are stronger than that of farther pairs, it is intuitive that the upper inputs of $\mathbf{V}$ should have more weightings than the lower ones in pre-filtering.

It turns out that for this kind of decreasing rotation angles, the simplified model from the forward elimination in Fig. 2(a) is a more robust approximation than the backward elimination in Fig. 2(b) since the forward model

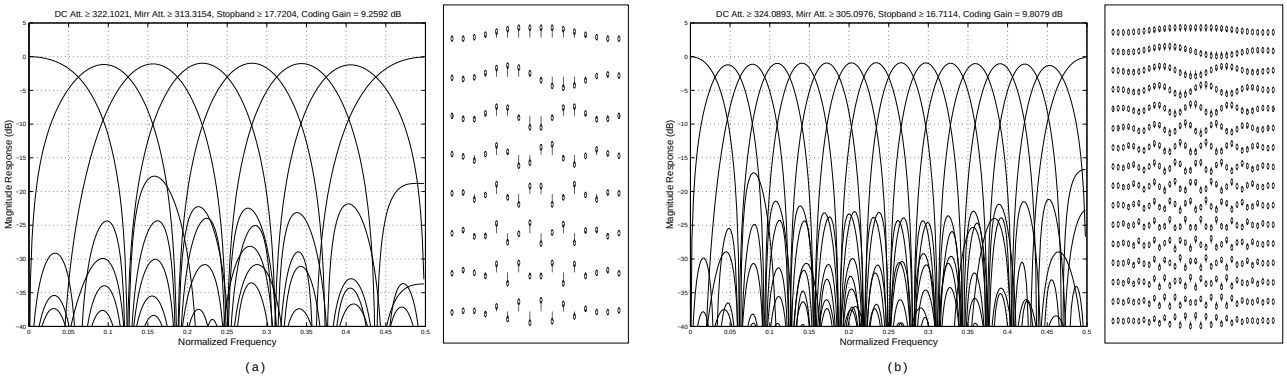


Fig. 3. Design examples of optimized TDLOT. (a) $8 \times 16$ TDLOT; coding gain 9.2592 dB. (b) $16 \times 32$ TDLOT; coding gain: 9.8079 dB.

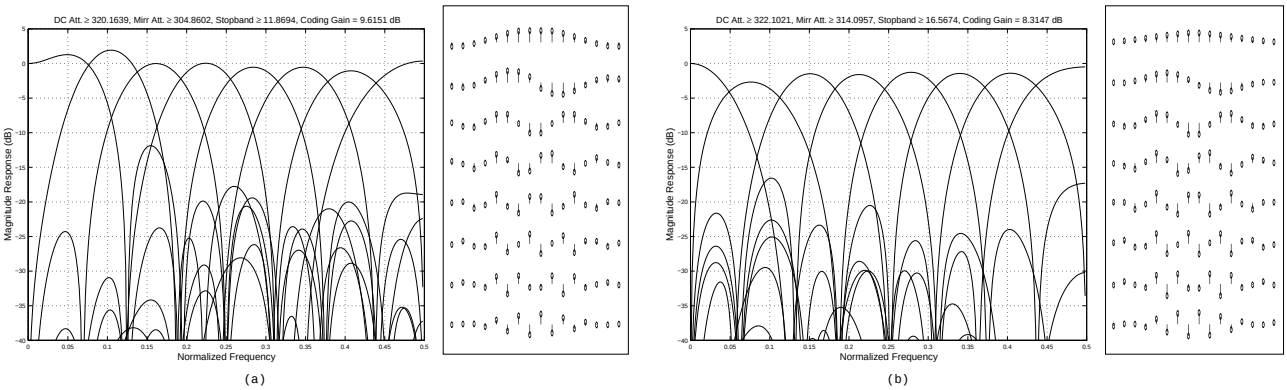


Fig. 4. Optimized $8 \times 16$ TDLT: (a) Analysis bank; coding gain: 9.6151 dB. (b) Synthesis bank; coding gain: 8.3147 dB.

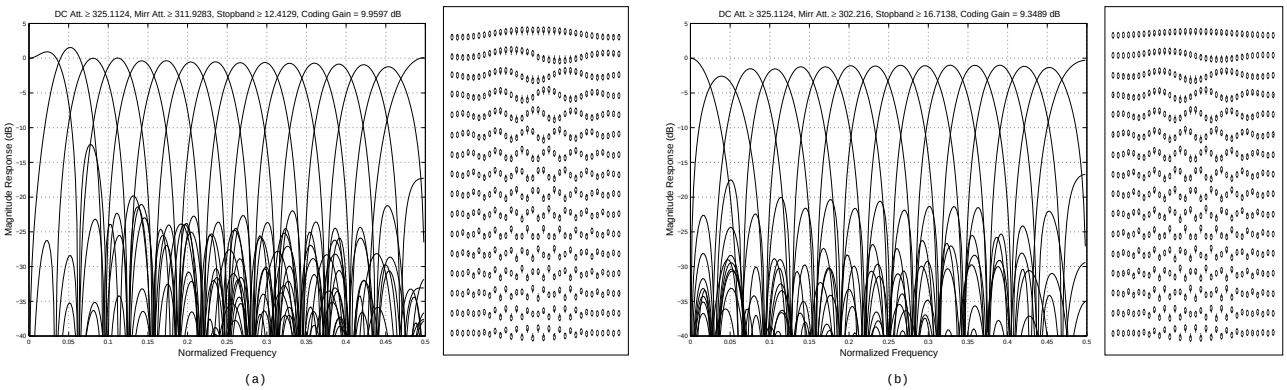


Fig. 5. Optimized $16 \times 32$ TDLT: (a) Analysis bank; coding gain: 9.9597dB. (b) Synthesis bank; coding gain: 9.3489 dB.

TABLE II

CODING GAIN IN DB OF VARIOUS FAST TDLTS FOR AN AR(1) SIGNAL WITH $\rho = 0.95$

| Size | Initial TDLOT-II | Optimized TDLOT-II | Initial TDLT-III (no scale) | Optimized TDLT-III (no scale) | Optimized TDLT-III (with scale) | Optimized TDLT-IV | Optimized TDLT-V |
|---|---|---|---|---|---|---|---|
| $4 \times 6$ | 7.5701 | 7.5701 | 7.5701 | 8.0685 | 8.0685 | 8.0685 | 8.0685 |
| $4 \times 8$ | 7.9259 | 7.9426 | 8.0761 | 8.1327 | 8.6349 | 8.6349 | 8.6349 |
| $8 \times 10$ | 8.8259 | 8.8259 | 8.8259 | 9.0611 | 9.0611 | 9.0611 | 9.0611 |
| $8 \times 12$ | 8.9910 | 9.0015 | 9.0610 | 9.0750 | 9.3375 | 9.3375 | 9.3375 |
| $8 \times 14$ | 9.1050 | 9.1380 | 9.2099 | 9.2539 | 9.5030 | 9.5022 | 9.5022 |
| $8 \times 16$ | 9.2306 | 9.2563 | 9.3200 | 9.3814 | 9.6115 | 9.6005 | 9.6005 |
| $16 \times 32$ | 9.7941 | 9.8014 | 9.7510 | 9.8298 | 9.9496 | 9.9057 | 9.9039 |
| $32 \times 64$ | 9.9965 | 10.0046 | 9.9563 | 9.9916 | 10.0563 | 9.9978 | 9.9822 |
| $64 \times 128$ | 10.0668 | 10.0737 | 10.0248 | 10.0545 | 10.0894 | 10.0453 | 9.9572 |

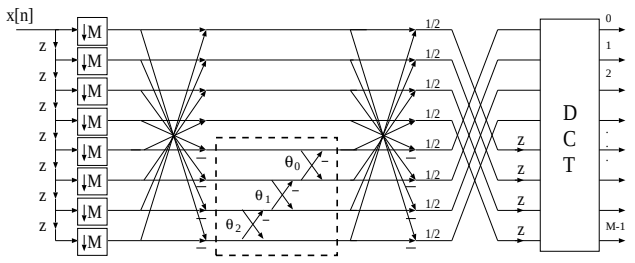


Fig. 6. Structure of the fast TDLOT-II.

TABLE III
OPTIMIZED ROTATION ANGLES FOR DIFFERENT PRE-FILTERS IN THE TDLOT-II

| Size of $\mathbf{V}$ | Rotation angles: $\theta_0$ to $\theta_{N-2}$ |
|---|---|
| $2 \times 2$ | $-0.10\,\pi$. |
| $3 \times 3$ | $-0.15\,\pi$, $-0.07\,\pi$ |
| $4 \times 4$ | $-0.17\,\pi$, $-0.12\,\pi$, $-0.05\,\pi$ |
| $5 \times 5$ | $-0.19\,\pi$, $-0.15\,\pi$, $-0.10\,\pi$, $-0.04\,\pi$ |
| $6 \times 6$ | $-0.20\,\pi$, $-0.17\,\pi$, $-0.13\,\pi$, $-0.09\,\pi$, $-0.04\,\pi$ |
| $7 \times 7$ | $-0.21\,\pi$, $-0.19\,\pi$, $-0.16\,\pi$, $-0.12\,\pi$, $-0.08\,\pi$, $-0.03\,\pi$ |
| $8 \times 8$ | $-0.21\,\pi$, $-0.20\,\pi$, $-0.18\,\pi$, $-0.15\,\pi$, $-0.11\,\pi$, $-0.07\,\pi$, $-0.03\,\pi$ |

starts from smaller angles, and yields less accumulation error.

It is worthwhile to compare the TDLOT-II with the LOT-I. In LOT-I, the post-processing are performed in the transform domain. It can be verified that in this case, the backward factorization-based simplified model yields better performance than the forward method. For $M = 8$, the main angles from the backward factorization of the matrix $\mathbf{V}$ in LOT-II are $[0.16\pi, 0.17\pi, 0.14\pi]$, and the optimized results given in [3], [6] are $[0.13\pi, 0.16\pi, 0.13\pi]$, corresponding to a coding gain of $9.1973\,dB$. In both cases, the principle angles are close to each others. This is quite different from the angles in the TDLOT-II. Therefore, the model in the LOT-I has more accumulation error than the TDLOT-II, which explains its limited performance when $M > 16$ [3].

For instance, when $M = 16$, the coding gain of the LOT-II is $9.7593\,dB$. After backward factorization and discarding trivial angles, the coding gain of the resulting LOT-I drops to $9.6826\,dB$. Compared with this, the simplified TDLOT-II in Fig. 2(a) produces a coding gain of $9.7941\,dB$, even higher than the original TDLOT-I. In short, the decreasing nature of the rotation angles in the TDLOT-II makes it a very robust model, even when $M > 16$, as verified by Table II.

## IV. Lifting-based fast algorithm

Although the simplified model in TDLOT-II is faster than the TDLT-I, it still involves floating multiplications, which are slow and undesired in many software and hardware applications. In this section, a lifting-based fast TDLT is developed, paving the path to much faster multiplierless solutions.

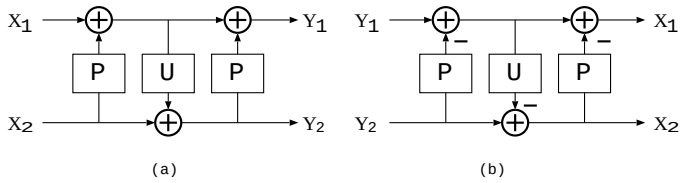


Fig. 7. Representation of a plane rotation by three lifting steps. (a) Forward representation. (b) Corresponding inverse.

The lifting scheme is essentially the elementary row operation in matrix theory. It is proposed in [19] as a tool for filter bank design. More systematic and general results were presented in [4], [5]. The lifting structure enables flexible, fast design and implementation of biorthogonal transforms. It also allows lossless transformation with reversible integer mapping.

Fig. 7(a) illustrates the decomposition of a plane rotation into three lifting steps [19], [4], [5]. This can be written in matrix form as

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 1 & P \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ U & 1 \end{bmatrix} \begin{bmatrix} 1 & P \\ 0 & 1 \end{bmatrix}, \tag{11}$$

where

$$\begin{aligned} P &= (\cos\theta - 1)/\sin\theta = -tan(\theta/2), \\ U &= \sin\theta. \end{aligned} \tag{12}$$

Each lifting step is a biorthogonal transform, and its inverse also has a simple lifting structure. As a result, the inverse of a plane rotation can be represented by lifting steps as

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}^{-1} = \begin{bmatrix} 1 & -P \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -U & 1 \end{bmatrix} \begin{bmatrix} 1 & -P \\ 0 & 1 \end{bmatrix}, \tag{13}$$

as shown in Fig. 7(b). To inverse a lifting step, we simply need to subtract out what was added at the forward transform. This means that the original signal can still be perfectly reconstructed even if the floating-point multiplication results in the lifting steps are rounded to integers, as long as the same procedure is applied to both the forward and the inverse. This is the basis for many lifting-based lossless transforms [2], [20], [21]. Another advantage of the lifting step over the butterfly operation is that it enables in-place computation, i.e., no buffer is required – a desired property in hardware implementations.

However, floating-point multiplications are still needed in the above approach. To obtain fast implementation, we can approximate the floating-point lifting coefficients by hardware-friendly dyadic values (*i.e.*, rationals in the format of $k/2^m$; $k$, $m$ are integers), which can be implemented by only shift and addition operations. In doing so, we can achieve a very fast approximation of the original transform. The elimination of the multiplication can also reduce the dynamic range of the transform [2].

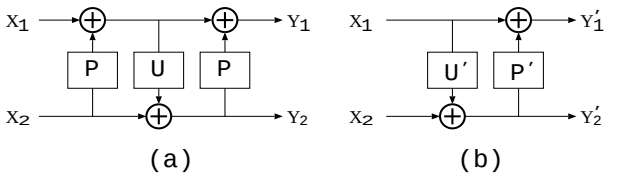


Fig. 8. Approximating a small rotation angle by two lifting steps. (a) Standard three-lifting representation. (b) Simplified two-lifting structure.

A trivial lifting-based pre-filter for the TDLT can be obtained from the TDLOT-II structure by replacing each rotation angle with its three-lifting representation in (11)-(12). However, notice that

$$\begin{aligned} \lim_{\theta\to 0} P &= -\lim_{\theta\to 0} \tan\frac{\theta}{2} = 0, \\ \lim_{\theta\to 0} U &= -\lim_{\theta\to 0} \sin\theta = 0. \end{aligned} \tag{14}$$

Since the rotation angles in the TDLOT-II are steadily decreasing, it can be seen from (14) that many lifting parameters would have small magnitudes. Hence, it is possible to achieve a close approximation of such a rotation angle by employing only two lifting steps as shown in Fig. 8(b). This can be justified by the following analysis.

In Fig. 8(a), the outputs of the three-lifting structure can be written as:

$$\begin{aligned} Y_1 &= (1 + P\,U)X_1 + (2\,P + P^2\,U)X_2, \\ Y_2 &= U\,X_1 + (1 + P\,U)X_2, \end{aligned} \tag{15}$$

whereas the outputs of the simplified structure in Fig. 8(b) are:

$$\begin{aligned} {Y_1}' &= (1 + P'\,U')X_1 + P'\,X_2, \\ {Y_2}' &= U'\,X_1 + X_2. \end{aligned} \tag{16}$$

If the rotation angle is small enough such that the magnitudes of its lifting parameters are much less than unity, all second order and third order terms in (15) and (16) can be ignored, and the following setting of the two-lifting model can approximate the standard model closely:

$$\begin{aligned} P' &= 2\,P, \\ U' &= U. \end{aligned} \tag{17}$$

When the rotation angle is not small enough and only the third order term can be ignored, the following choice will yield a more accurate approximation:

$$\begin{aligned} P' &= 2\,P, \\ U' &= U/2. \end{aligned} \tag{18}$$

By replacing the rotation angles in the TDLOT-II with the two-lifting structure, we can obtain another simplified model for the matrix $\mathbf{V}$ as illustrated in Fig. 9(a). The transform can be made even faster if each lifting parameter is approximated by an appropriate dyadic value. This is similar to the approach taken in the LiftLT design [7].

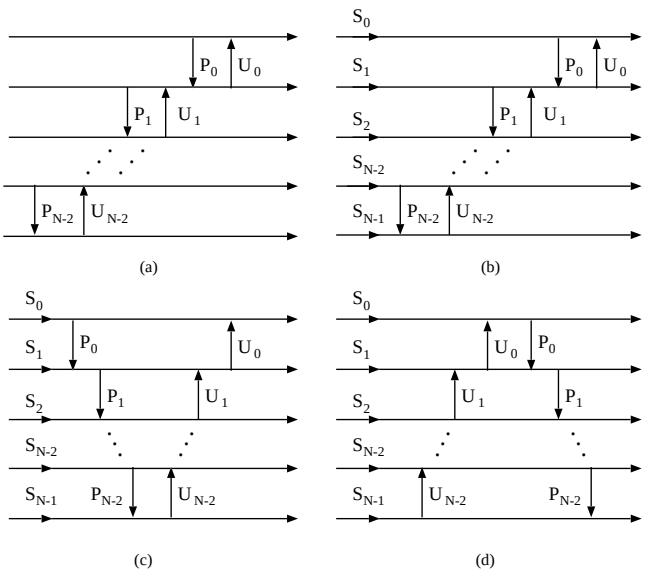


Fig. 9. Simplified structures for the matrix $\mathbf{V}$ in the pre-filter of the TDLT. (a) Structure that approximates the cascading of rotation angles. (b) Structure in the TDLT-III. (c) Structure in the TDLT-IV. (d) Structure in the TDLT-V.

TABLE IV
EXAMPLES OF OPTIMIZED PARAMETERS FOR DIFFERENT PRE-FILTERS IN TDLT-III

| Size of $\mathbf{V}$ | $S_0 \ldots S_{N-1}$ | $\{P_0, U_0\} \ldots \{P_{N-2}, U_{N-2}\}$ |
|---|---|---|
| $1 \times 1$ | 1.55 | - |
| $2 \times 2$ | 1.39, 1.29 | -0.17, 0.57 |
| $3 \times 3$ | 1.39, 1.15, 1.22 | -0.33, 0.71, -0.13, 0.41 |
| $4 \times 4$ | 1.40, 1.12, 1.14, 1.19 | -0.40, 0.78, -0.33, 0.56, -0.11, 0.34 |
| $5 \times 5$ | 1.40, 1.10, 1.09, 1.11, 1.15 | -0.43, 0.81, -0.44, 0.64, -0.30, 0.50, -0.10, 0.30 |
| $6 \times 6$ | 1.40, 1.10, 1.08, 1.08, 1.10, 1.13 | -0.45, 0.84, -0.50, 0.69, -0.41, 0.58, -0.28, 0.45, -0.10, 0.26 |
| $7 \times 7$ | 1.41, 1.09, 1.07, 1.07, 1.07, 1.09, 1.12 | -0.47, 0.85, -0.54, 0.72, -0.48, 0.64, -0.38, 0.54, -0.26, 0.41, -0.09, 0.23 |
| $8 \times 8$ | 1.41, 1.09, 1.06, 1.06, 1.06, 1.06, 1.08, 1.11 | -0.47, 0.86, -0.56, 0.74, -0.53, 0.67, -0.46, 0.60, -0.36, 0.50, -0.24, 0.37, -0.09, 0.21 |

The structure in Fig. 9(a) is designed as a close approximation of the orthogonal TDLOT-II, whose coding gain is not as high as the biorthogonal case. However, by introducing a scaling coefficient to each channel of the matrix $\mathbf{V}$ as shown in Fig. 9(b), the structure would approximate the SVD model very well. The corresponding biorthogonal TDLT is given in Fig. 10, denoted as the TDLT-III. The butterflies in the TDLT-III are also implemented by lifting steps as discussed in [1]. For a $M \times (M + 2N)$ TDLT-III, $\mathbf{V}$ has only $3N - 2$ parameters, representing a dramatic simplification over the $N^2$ parameters of the SVD model.

The coding gain results of various TDLT-III versions can be found in Table II. The frequency responses and impulse responses of several optimized TDLT-III are shown in Fig. 11–13. In addition, Table IV lists the optimized parameters for $\mathbf{V}$ of different sizes. In Table II, the initial TDLT-III

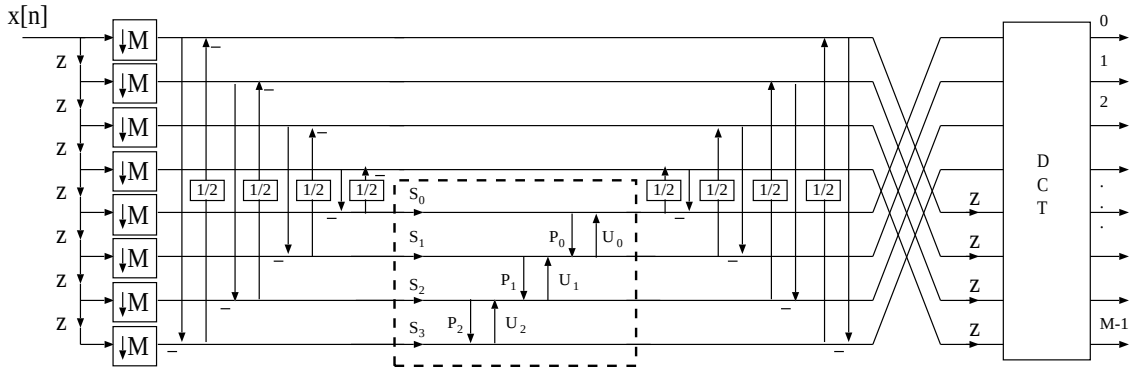


Fig. 10. General structure of the TDLT-III.

are obtained from (18) when $M \leq 16$, and from (17) when $M > 16$. The optimized TDLT-III results from further optimizing the lifting and scaling parameters in Fig. 10.

Table II reveals that the coding gains of the analysis banks of the TDLT-III without scalings approximate their TDLOT-II originals very well. The optimized results of the TDLT-III with scalings have much higher coding gains. In fact, the result is virtually identical to the optimized SVD-based TDLT in Table I when $M \leq 16$. For the most popular $8 \times 16$ case, the optimized TDLT-III yields $9.6115\,dB$, a negligible loss comparing to the $9.6151\,dB$ of the SVD-based optimized TDLT. This result is close to the $9.6131\,dB$ of the GLT in [23]. However, the GLT is based on the LOT-II and it is much more computational expensive than the TDLT-III.

It is interesting to note that for $N = 2$, this model for $\mathbf{V}$ reduces to the LDU factorization of a matrix [16], which is equivalent to the SVD. Therefore the model in TDLT-III is a complete model for all invertible $2 \times 2$ matrices.

## V. Alternative models for the TDLT-III

Besides the SVD decomposition, the LU factorization [16] provides another model for the invertible matrix $\mathbf{V}$ in the TDLT. With elementary row and column operations, any $N \times N$ invertible matrix $\mathbf{A}$ can be written as:

$$\mathbf{A} = \mathbf{PLDU}, \tag{19}$$

where $\mathbf{P}$ is a permutation matrix, $\mathbf{L}$ (respectively $\mathbf{U}$) is lower (respectively upper) triangular with all diagonal entries equal to 1, and $\mathbf{D}$ is a nonsingular diagonal matrix. Similarly, $\mathbf{A}$ can also be written as $\mathbf{PUDL}$. In fact, the position of the diagonal matrix $\mathbf{D}$ in the LU factorization is quite flexible. As a result, we can also write $\mathbf{A}$ as $\mathbf{PDLU}$ or $\mathbf{PLUD}$. Each of $\mathbf{L}$ and $\mathbf{U}$ can be parameterized by $N(N-1)/2$ elementary row operators (lifting steps). Hence the total number of parameters is $N^2$, which can not be used directly for fast implementation.

Two simplified models that resemble the LU factorization and provide good performance in the TDLT framework are given in Fig. 9(c) and Fig. 9(d). We denote the corresponding TDLT as TDLT-IV and TDLT-V. They have the same complexity as the TDLT-III, *i.e.*, $N$ scalings and $2(N-1)$ lifting steps. The scaling coefficients can be placed between the upper and lower triangular parts or at the end of the signal flow without losing any optimal performance.

The structure of $\mathbf{V}$ in the TDLT-IV is closely related to that of the TDLT-III. In particular, when $N < 8$, the pre-filter parameters for TDLT-III can be directly ported to TDLT-IV and achieve very similar performance. This can be seen from Fig. 14, where the output of the matrix $\mathbf{V}$ in the TDLT-III can be written as:

$$\begin{cases} Y_{N-1} = S_{N-1}X_{N-1} + P_{N-2}S_{N-2}X_{N-2}, \\ Y_k = S_kX_k + P_{k-1}S_{k-1}X_{k-1} + U_kY_{k+1}, \quad k = N-2, \ldots, 0. \end{cases} \tag{20}$$

From Fig. 14(b), we have $X'_0 = S_0X_0$ and the following approximations:

$$\begin{aligned} X'_k &= S_kX_k + P_{k-1}S_{k-1}X_{k-1} + \ldots + P_{k-1}\ldots P_1P_0S_0X_0 \\ &\approx S_kX_k + P_{k-1}S_{k-1}X_{k-1} \end{aligned} \tag{21}$$

for $k = 1, \ldots, N-1$, since all the lifting parameters are less than 1 and many of them are actually very small. This approximation holds true when $N < 8$. In this case, the outputs can be written as

$$\begin{aligned} Y'_{N-1} &= X'_{N-1} \\ &\approx S_{N-1}X_{N-1} + P_{N-2}S_{N-2}X_{N-2} = Y_{N-1}, \end{aligned} \tag{22}$$

and

$$\begin{aligned} Y'_k &= X'_k + U_kY'_{k+1} \\ &\approx S_kX_k + P_{k-1}S_{k-1}X_{k-1} + U_kY_{k+1} = Y_k, \end{aligned} \tag{23}$$

for $k = N-2, \ldots, 0$. This proves that the TDLT-III and TDLT-IV would have similar performance if they use the same parameters.

Several optimized results for these two models are also given in Table II. For $M < 16$, their performances are very similar to the TDLT-III. For $2 \times 2$ matrix $\mathbf{V}$, these models also reduce to the true LU factorization.

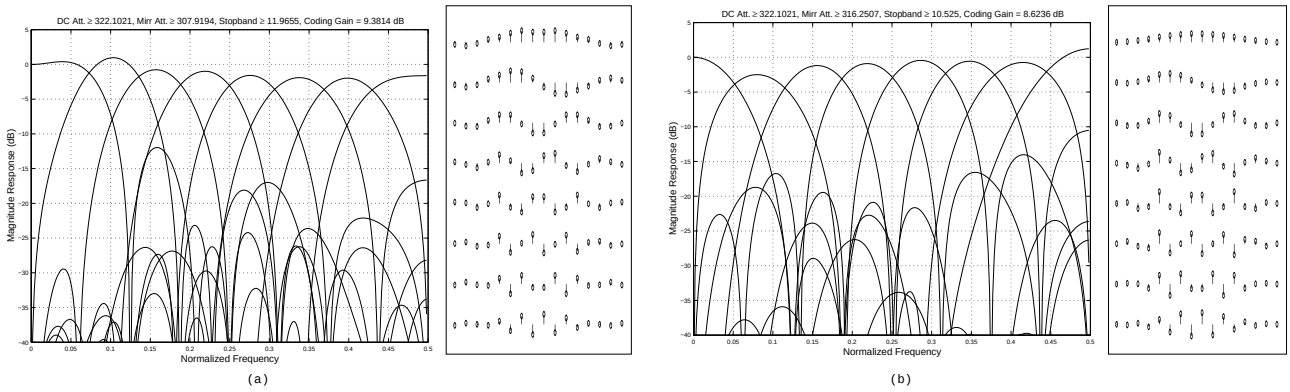


Fig. 11. Optimized $8 \times 16$ TDLT-III without scaling; Coding gain: 9.3814 dB.

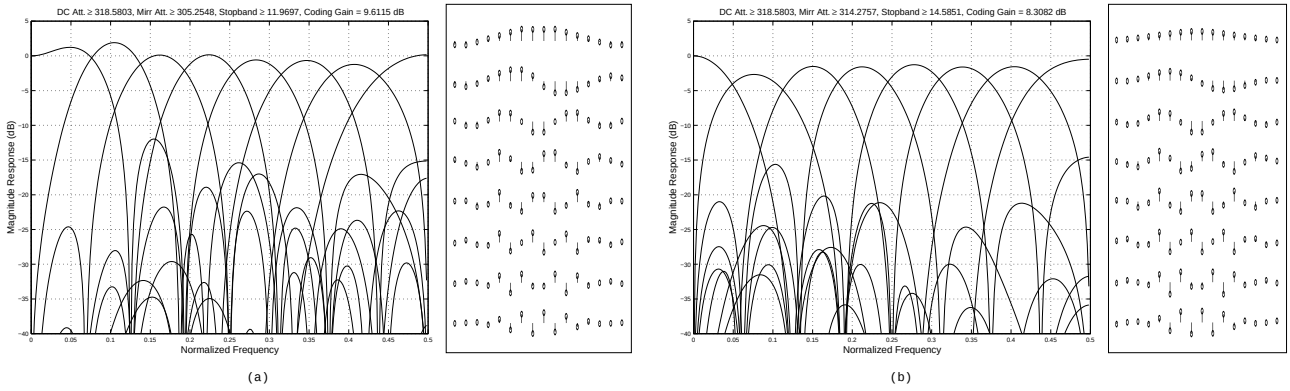


Fig. 12. Optimized $8 \times 16$ TDLT-III with scaling; Coding gain: 9.6115 dB.

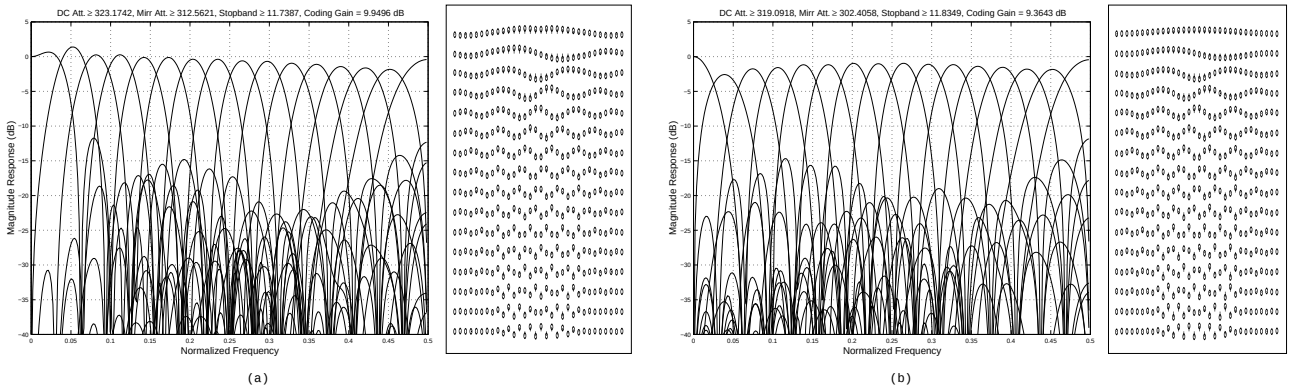


Fig. 13. Optimized $16 \times 32$ TDLT-III with scaling; Coding gain: 9.9496 dB.

## VI. Odd-Channel Fast TDLT

The framework of the TDLT makes the design of odd-channel lapped transform as convenient as that of the even-channel case. Odd-channel TDLT of different lengths can be easily obtained by changing the number of borrowing pixels at the boundary of neighboring blocks. The only difference in the odd-channel case is that for a given odd block size $M$, the maximum size of the TDLT is $M \times (2M - 1)$.

The optimized coding gain results for some odd-channel TDLTs are tabulated in Table V. An example of $5 \times 9$ TDLOT-II is given in Fig. 15, which has three symmetric filters and two anti-symmetric filters. Experimental results also reveal that the optimized matrix $\mathbf{V}$ in the odd-channel TDLT is identical to $\mathbf{V}$ of the same size in the even-channel case. In other words, the behavior of the optimized pre-filter is only determined by its own size, not the size $M$ of the signal block.

## VII. Computational complexity Analysis

In this section, we summarize the computational complexity of various fast TDLTs developed in this paper when floating-point implementation is considered.

The fast DCT algorithms in [8], [9], [13] is used to mea-

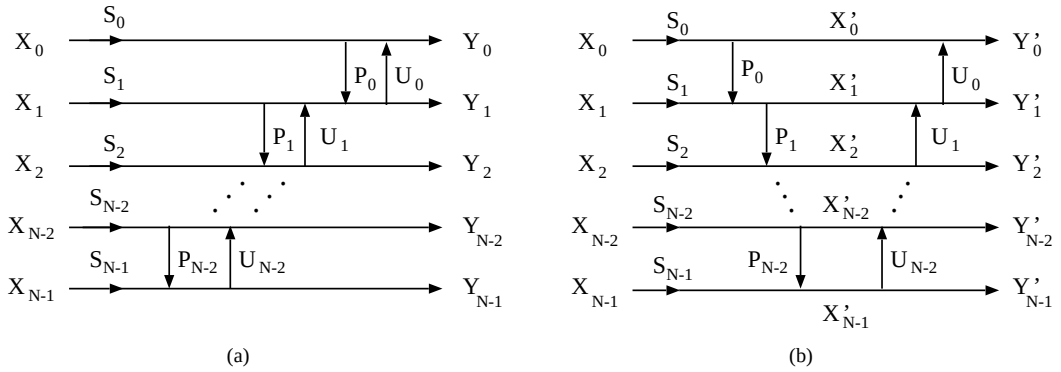


Fig. 14. Relationship between TDLT-III and TDLT-IV: (a) $\mathbf{V}$ in TDLT-III; (b) $\mathbf{V}$ in TDLT-IV.

TABLE V
CODING GAINS IN DB OF VARIOUS FAST ODD-CHANNEL TDLTS FOR AN AR(1) SIGNAL WITH $\rho = 0.95$

| Size of TDLT | Opt. TDLOT-II | Opt. TDLT-III |
|---|---|---|
| $5 \times 9$ | 8.3625 | 8.9144 |
| $9 \times 13$ | 9.1209 | 9.4177 |
| $9 \times 17$ | 9.3394 | 9.6619 |
| $15 \times 23$ | 9.6298 | 9.8243 |
| $15 \times 29$ | 9.7501 | 9.9216 |
| $31 \times 47$ | 9.9328 | 10.0153 |
| $31 \times 61$ | 9.9944 | 10.0508 |

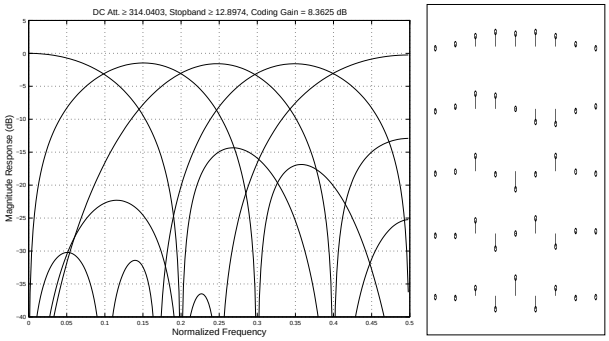


Fig. 15. Optimized $5 \times 9$ TDLOT-II. Coding gain: $8.3625\,dB$.

sure the complexity of the TDLT. Define $\mu(M)$ and $\alpha(M)$ as the number of floating-point multiplications and additions required by a $M$-channel transform, respectively. The computational complexity of this kind of DCT algorithm for even $M$ is given by:

$$\begin{aligned} \mu_C(M) &= \frac{M}{2} log_2 M + 1, \\ \alpha_C(M) &= \frac{3M}{2} log_2 M - M + 1. \end{aligned} \tag{24}$$

Each rotation angle in the TDLOT-II can be implemented with 3 multiplications and 3 additions [13], thus the complexity of a $M \times (M + 2N)$ TDLOT-II is:

$$\begin{aligned} \mu_{II}(M,\, 2N) &= \mu_C(M) + (3N - 3), \\ \alpha_{II}(M,\, 2N) &= \alpha_C(M) + (7N - 3). \end{aligned} \tag{25}$$

TABLE VI
COMPUTATIONAL COMPLEXITY OF DCT AND FAST $M \times 2M$ ($M$ IS EVEN) TDLTS

| | Number of MULs | | | Number of ADDs | | |
|---|---|---|---|---|---|---|
| M | DCT | TDLOT -II | TDLT -III | DCT | TDLOT -II | TDLT -III |
| 2 | 2 | 2 | 3 | 2 | 2 | 6 |
| 4 | 5 | 8 | 9 | 9 | 20 | 19 |
| 8 | 13 | 22 | 23 | 29 | 54 | 51 |
| 16 | 33 | 54 | 55 | 81 | 134 | 127 |
| 32 | 81 | 126 | 127 | 209 | 318 | 303 |
| 64 | 193 | 286 | 287 | 513 | 734 | 703 |
| 128 | 449 | 638 | 639 | 1217 | 1662 | 1599 |

The complexity of the lifting-based $M \times (M+2N)$ TDLT-III, TDLT-IV and TDLT-V is given by:

$$\begin{aligned} \mu_{III}(M,\, 2N) &= \mu_C(M) + (3N - 2), \\ \alpha_{III}(M,\, 2N) &= \alpha_C(M) + (6N - 2). \end{aligned} \tag{26}$$

The $\frac{1}{2}$ normalization of the butterflies is not counted in the above formula. Table VI compares the complexity of the DCT versus those of various fast TDLTs with full borrowing. The TDLOT-II and TDLT-III still need 40% to 70% overhead above the DCT. However, they lead to much improved coding performances.

## VIII. TDLT WITH RATIONAL AND DYADIC COEFFICIENTS

This section investigates the approximation of the optimized values for the free parameters in the TDLT-III and TDLT-IV by various rational and dyadic values. These designs lead to fast, sometimes even multiplierless implementations, and also allow for lossless compression.

The fast DCT algorithm chosen for the following examples is the lifting-based binDCT [2], [14], which is derived from the well-known Chen-Wang factorization of the DCT [8], [9], [13]. For $M = 8$, the structures of the DCT factorization and the binDCT are given in Fig. 16. This binDCT example needs 27 shifts and 37 additions, and has a coding gain of $8.8220\,dB$ (the DCT has $8.8259\,dB$). The final scalings of the binDCT should be combined with the quantization to reduce the complexity even further.

TABLE VII
EXAMPLES OF FAST TDLT-III AND TDLT-IV WITH RATIONAL OR DYADIC PARAMETERS

| Size of TDLT | Cfg. | $S_0$ | $S_1$ | $S_2$ | $S_3$ | $P_0$ | $U_0$ | $P_1$ | $U_1$ | $P_2$ | $U_2$ | # of Muls. | # of Shifts | # of Adds. | Cod. Gain TDLT-III | Cod. Gain TDLT-IV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $8 \times 10$ | 1 | 3/2 | - | - | - | - | - | - | - | - | - | 0.5 | 25.5 | 41.5 | 9.0529 | 9.0529 |
| $8 \times 12$ | 1 | 4/3 | 4/3 | - | - | -3/16 | 1/2 | - | - | - | - | 1 | 31 | 50 | 9.3173 | 9.3173 |
| | 2 | 3/2 | 3/2 | - | - | -3/16 | 1/2 | - | - | - | - | 1 | 31 | 50 | 9.2710 | 9.2710 |
| | 3 | 1 | 1 | - | - | -3/16 | 1/2 | - | - | - | - | 0 | 30 | 48 | 9.0670 | 9.0670 |
| $8 \times 14$ | 1 | 3/2 | 5/4 | 5/4 | - | -1/8 | 3/8 | -3/8 | 3/4 | - | - | 1.5 | 37.5 | 59 | 9.4740 | 9.4702 |
| | 2 | 4/3 | 8/7 | 8/7 | - | -1/8 | 3/8 | -3/8 | 3/4 | - | - | 1.5 | 37.5 | 59 | 9.4758 | 9.4746 |
| | 3 | 1 | 1 | 1 | - | 0 | 3/8 | -3/8 | 11/16 | - | - | 0 | 33 | 56 | 9.2462 | 9.2462 |
| $8 \times 16$ | 1 | 4/3 | 8/7 | 8/7 | 8/7 | -1/16 | 1/4 | -1/4 | 1/2 | -3/8 | 3/4 | 2 | 40 | 65 | 9.5927 | 9.5845 |
| | 2 | 3/2 | 9/8 | 9/8 | 9/8 | -1/16 | 1/4 | -1/4 | 1/2 | -3/8 | 3/4 | 2 | 40 | 65 | 9.5848 | 9.5730 |
| | 3 | 4/3 | 1 | 1 | 1 | 0 | 1/4 | -1/4 | 1/2 | -1/4 | 3/4 | 0.5 | 36.5 | 60 | 9.4929 | 9.4880 |
| | 4 | 3/2 | 1 | 1 | 1 | 0 | 1/4 | -1/4 | 1/2 | -1/4 | 3/4 | 0.5 | 36.5 | 60 | 9.4718 | 9.4651 |
| | 5 | 1 | 1 | 1 | 1 | 0 | 1/4 | -1/4 | 1/2 | -1/2 | 3/4 | 0 | 36 | 59 | 9.3676 | 9.3548 |

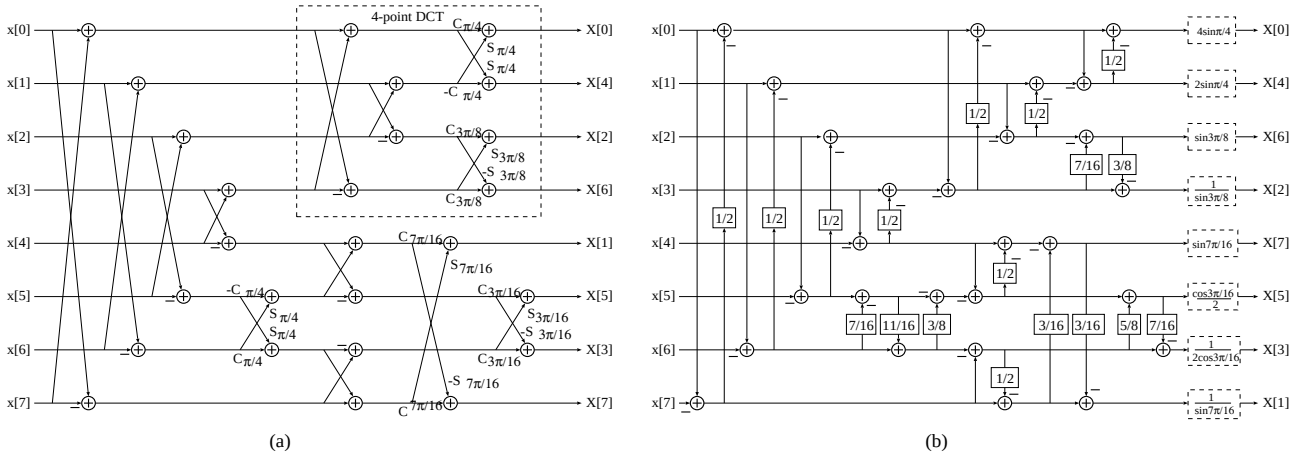


Fig. 16. DCT versus binDCT. (a) Chen-Wang algorithm of the 8-point DCT. (b) The corresponding 8-point binDCT.

Table VII tabulates various rational approximations for $\mathbf{V}$ in the $8 \times 10$, $8 \times 12$, $8 \times 14$, and $8 \times 16$ TDLT-III and TDLT-IV. The binDCT example in Fig. 16(b) is used in measuring the overall complexity and the coding gain. Compared to the results in Table II, the performance loss due to finite-length approximation of the optimized floating parameters and the binDCT is negligible. For instance, the first two $8 \times 16$ TDLT-III examples have coding gains of 9.5927 and 9.5848 $dB$, respectively, which are almost identical to the 9.6115 $dB$ obtained from floating-point parameters. The results in Table VII also show that with the same parameters, the performance of the TDLT-IV is very close to that of the TDLT-III for $M = 8$. This verifies the analysis in Section V.

Reversible integer-to-integer mapping, a critical requirement for lossless compression, can be easily achieved by setting all scaling factors in the matrix $\mathbf{V}$ to be unity. In this case, the transform can be implemented with only shift and addition operations, simplifying the hardware implementation significantly. However, the coding gain is sacrificed by about 0.2 $dB$. Nevertheless, it still has higher coding gain and much faster implementation than the lossless LOT in [21], which has similar coding gain to the LOT-I and still employs floating operations.

As an example of dyadic TDLT, the filter coefficients of the $8 \times 16$ TDLT-III of Configuration 5 in Table VII are tabulated in Table VIII, without taking into consideration the final scaling of the binDCT.

The approximation of various optimized $\mathbf{V}$ matrices can be used in TDLT of other sizes. For example, a $4 \times 8$ TDLT-III with dyadic coefficients is given in Fig. 17, where the 4-point binDCT is used, and the parameters for $\mathbf{V}$ are chosen from Table VIII. Its coding gain is 8.1319 $dB$, much better than the 7.5701 $dB$ of the 4-point DCT. The dyadic coefficients of this TDLT are given in Table IX.

## IX. Experimental Results

The performance of various binDCT-based fast TDLT-III in image coding is demonstrated in this section. The SPIHT algorithm [24] is used in the coding of coefficients produced by all transforms. In other words, the coding algorithm is fixed; we simply compare the effectiveness of various signal decomposition schemes. The benchmark transformation is the popular 9/7-tap biorthogonal wavelet with irrational coefficients.

The DCT/TDLT transform coefficients are rearranged in analogy to the wavelet representation [25], [26]. As many as possible levels of 9/7 wavelet decomposition is employed to decorrelate the DC subband of the DCT/TDLT coefficients [25]. This ensures that the SPIHT coding algorithm has the same tree depth in all cases. To generate the DCT results,

TABLE VIII

THE FIRST PART OF THE ANALYSIS FILTERS $\{h_i\}$ AND SYNTHESIS FILTERS $\{g_i\}$ CORRESPONDING TO THE $8 \times 16$ TDLT-III CFG. 5 IN TABLE VII. THE SECOND PART CAN BE OBTAINED BY $h_i(n) = h_i(15-n)$, AND $g_i(n) = -g_i(15-n)$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $h_0$ | -15/$2^9$ | -7/$2^7$ | -9/$2^9$ | 21/$2^9$ | 43/$2^9$ | 73/$2^9$ | 23/$2^7$ | 79/$2^9$ |
| $h_1$ | -168939/$2^{21}$ | -100235/$2^{19}$ | -210453/$2^{21}$ | 300705/$2^{21}$ | 711007/$2^{21}$ | 1055429/$2^{21}$ | 237643/$2^{19}$ | 365547/$2^{21}$ |
| $h_2$ | -129/$2^{14}$ | -513/$2^{12}$ | -2439/$2^{14}$ | 1539/$2^{14}$ | 5309/$2^{14}$ | 5511/$2^{14}$ | -255/$2^{12}$ | -6719/$2^{14}$ |
| $h_3$ | 5347/$2^{16}$ | -1405/$2^{14}$ | -23203/$2^{16}$ | 4215/$2^{16}$ | 43401/$2^{16}$ | 9971/$2^{16}$ | -12099/$2^{14}$ | -34019/$2^{16}$ |
| $h_4$ | 9/$2^8$ | 1/$2^6$ | -33/$2^8$ | -3/$2^8$ | 67/$2^8$ | -31/$2^8$ | -17/$2^6$ | 55/$2^8$ |
| $h_5$ | 11/$2^{12}$ | 171/$2^{10}$ | -971/$2^{12}$ | -513/$2^{12}$ | 3073/$2^{12}$ | -3685/$2^{12}$ | 149/$2^{10}$ | 4085/$2^{12}$ |
| $h_6$ | -85/$2^{11}$ | 43/$2^9$ | -83/$2^{11}$ | -129/$2^{11}$ | 577/$2^{11}$ | -941/$2^{11}$ | 213/$2^9$ | -363/$2^{11}$ |
| $h_7$ | -5383/$2^{17}$ | 1753/$2^{15}$ | 775/$2^{17}$ | -5259/$2^{17}$ | 17547/$2^{17}$ | -38039/$2^{17}$ | 12519/$2^{15}$ | -60153/$2^{17}$ |
| $g_0$ | 0 | 13/$2^8$ | 13/$2^6$ | 3/$2^3$ | 5/$2^3$ | 51/$2^6$ | 243/$2^8$ | 1 |
| $g_1$ | 0 | -271/$2^{13}$ | 79/$2^{11}$ | 79/$2^{11}$ | 669/$2^{12}$ | 813/$2^{11}$ | 325/$2^{10}$ | 3/$2^5$ |
| $g_2$ | 0 | -977/$2^{13}$ | -241/$2^{11}$ | 21/$2^8$ | 107/$2^8$ | 689/$2^{11}$ | -815/$2^{13}$ | -1/2 |
| $g_3$ | 0 | -43/$2^{10}$ | -33/$2^8$ | -15/$2^9$ | 143/$2^9$ | 13/$2^8$ | -31/$2^7$ | -5/$2^5$ |
| $g_4$ | 0 | 27/$2^9$ | -37/$2^7$ | -3/$2^4$ | 11/$2^4$ | -27/$2^7$ | -283/$2^9$ | 1/2 |
| $g_5$ | 0 | 643/$2^{14}$ | -151/$2^{12}$ | -633/$2^{13}$ | 1529/$2^{13}$ | -693/$2^{12}$ | 23/$2^{11}$ | 93/$2^9$ |
| $g_6$ | 0 | 2061/$2^{16}$ | 173/$2^{14}$ | -321/$2^{11}$ | 705/$2^{11}$ | -7021/$2^{14}$ | 25331/$2^{16}$ | -3/$2^4$ |
| $g_7$ | 0 | -2093/$2^{17}$ | 1517/$2^{15}$ | -6441/$2^{16}$ | 12585/$2^{16}$ | -10105/$2^{15}$ | 6863/$2^{14}$ | -247/$2^9$ |

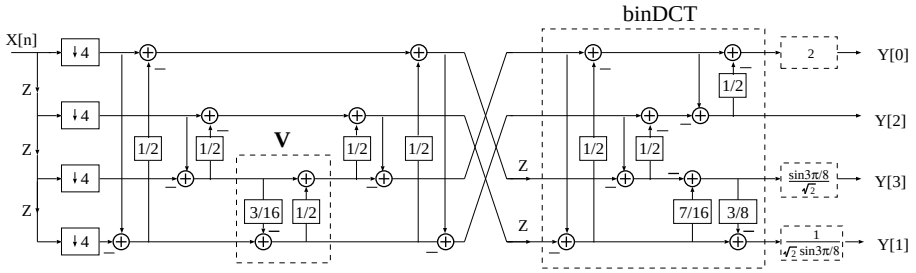


Fig. 17. A $4 \times 8$ TDLT with dyadic coefficients. The complexity is 14 shifts and 21 additions per 4 coefficients. Coding gain: 8.1319 $dB$.

.

TABLE IX

DYADIC COEFFICIENTS OF THE $4 \times 8$ TDLT IN FIG. 17

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $h_0$ | -1/16 | 9/256 | 55/256 | 5/16 | 5/16 | 55/256 | 9/256 | -1/16 |
| $h_1$ | -107/512 | 609/8192 | 6239/8192 | 299/512 | -299/512 | -6239/8192 | -609/8192 | 107/512 |
| $h_2$ | -1/8 | -3/128 | 67/128 | -3/8 | -3/8 | 67/128 | -3/128 | -1/8 |
| $h_3$ | -7/64 | -75/1024 | 523/1024 | -57/64 | 57/64 | -523/1024 | 75/1024 | 7/64 |
| $g_0$ | -3/64 | 1/4 | 3/4 | 67/64 | 67/64 | 3/4 | 1/4 | -3/64 |
| $g_1$ | -75/2048 | 7/128 | 57/128 | 523/2048 | -523/2048 | -57/128 | -7/128 | 75/2048 |
| $g_2$ | -9/128 | -1/8 | 5/8 | -55/128 | -55/128 | 5/8 | -1/8 | -9/128 |
| $g_3$ | -609/16384 | -107/1024 | 299/1024 | -6239/16384 | 6239/16384 | -299/1024 | 107/1024 | 609/16384 |

we simply turn pre- and post-filtering off.

The objective coding results (PSNR in dB) are tabulated in Table X. All test images are gray-scale 8-bit: $512 \times 512$ Lena, Goldhill and Barbara; $2048 \times 2560$ JPEG2000 test images – Cafe and Woman; and the first luminance frame of QCIF-size ($176 \times 144$) video sequences News and Glasgow.

Table X illustrates that simple pre/post-processing with DCT works well at all bit rates on images of different characteristics and resolutions. It can be seen from Table X that although the performance of the TDLT is not as good as the wavelet transform for smooth images Lena and Woman (where the input to the matrix $\mathbf{V}$ is quite small in many cases), it outperforms wavelet on the remaining. In particular, the improvement in Barbara can be as high as 1.5 $dB$ with the $8 \times 16$ TDLT-III. For low-resolution images, the TDLT also outperforms the 9/7 wavelet by a large margin. Even the $8 \times 12$ TDLT-III without scaling can outperform wavelet in some cases. On average, there

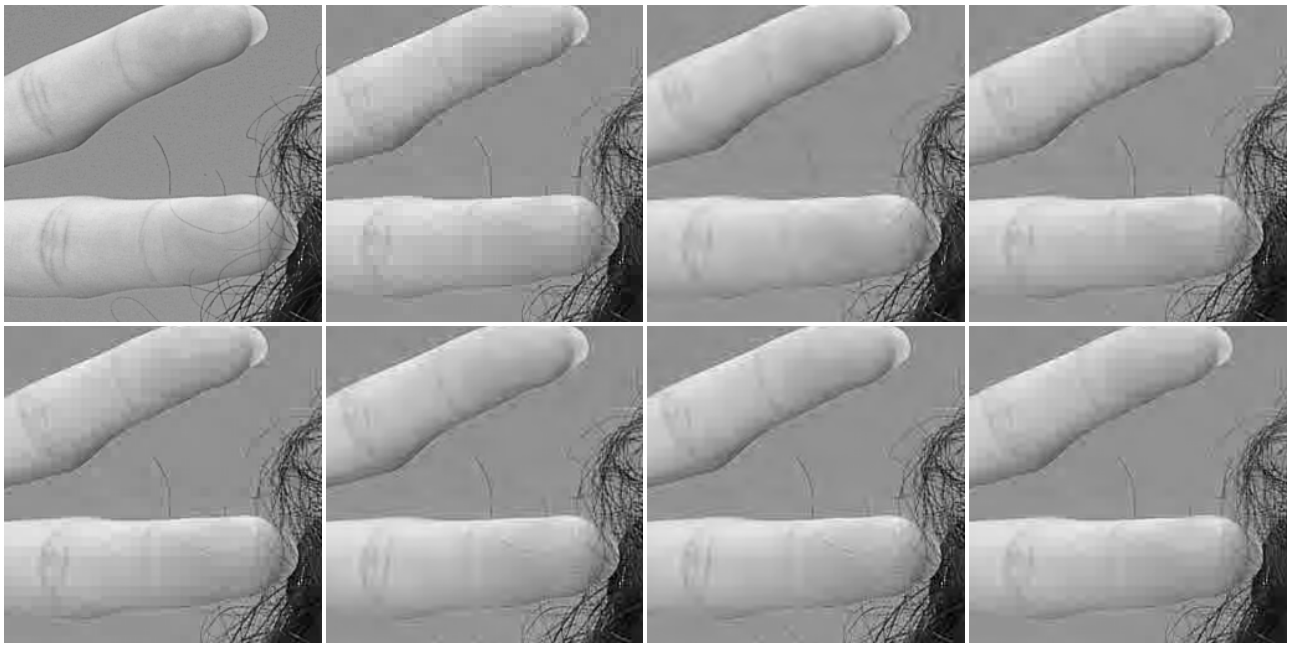

Fig. 18. Enlarged $256 \times 256$ Woman portions at $32:1$ compression ratio. Top row, from left to right: original image; coded by DCT (no pre/post-filtering): 29.53 dB; coded by 9/7 wavelet: 29.95 dB; coded by $8 \times 12$ TDLT Cfg. 1 (4-point pre/post-filtering): 29.84 dB; Bottom row, from left to right: coded by $8 \times 12$ TDLT Cfg. 3: 29.59 dB; coded by $8 \times 16$ TDLT Cfg. 1 (8-point pre/post-filtering): 29.89 dB; coded by $8 \times 16$ TDLT Cfg. 3: 29.80 dB; coded by DCT (no pre-filtering) and decoded by $8 \times 12$ TDLT Cfg. 1 (4-point post-filtering): 28.75 dB.

are about $0.15\,dB$ and $0.2\,dB$ drops when the $8 \times 16$ and $8 \times 12$ TDLT-III without scaling are used. However, they allow faster hardware and software implementations, and also enable lossless integer mapping.

Portions of reconstructed Woman images at 0.25 bits/pixel (bpp) are shown in Fig. 18. The TDLT effectively eliminates blocking artifacts and has similar visual quality to the wavelet-based results. The right-most image in the bottom row is encoded with the DCT and decoded by the $8 \times 12$ TDLT. The experiment intends to demonstrate the flexibility of the proposed TDLT framework. This can be viewed as the traditional post-filtering approach (without pre-filtering compensation), which has been proven to be quite effective in de-blocking.

## X. Conclusion

In this paper, we first present the optimized orthogonal and biorthogonal pre/post-filters for the time domain pre- and post-processed lapped transform (TDLT). Several simplified models are then developed to approximate the optimal pre/post-filters in the TDLT, based on cascades of either plane rotation operators or lifting steps. The same pre/post-filters can be used with the DCT of odd block sizes to generate various odd-channel TDLTs. These simplified models are found to be close and robust approximation of the optimal results, and they enable very fast implementations. Moreover, reversible integer mapping, which is critical to transform-based lossless compression, can be easily achieved with the lifting-based model when it is combined with lossless DCT implementations.

## References


[1] T. D. Tran, "Lapped transform via time-domain pre- and post-filtering. part I: general framework," *submitted to IEEE Trans. on Signal Processing*, Oct. 2001.

[2] Jie Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Trans. on Signal Processing, to appear*, Dec. 2001.

[3] H. S. Malvar, *Signal Processing with Lapped Transforms*, Norwood, MA: Artech House, 1992.

[4] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal*, vol. 3, pp. 186–200, 1996.

[5] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting step," *Journal Fourier Anal. Appl.*, vol. 4, pp. 247–269, 1998.

[6] H. S. Malvar and D. H. Staelin, "The LOT: transform coding without blocking effects," *IEEE Trans. on Signal Processing*, vol. 37, pp. 553–559, Apr. 1989.

[7] T. D. Tran, "The LiftLT: fast lapped transforms via lifting steps," *IEEE Signal Processing Letters*, vol. 7, pp. 145–149, Jun. 2000.

[8] W. Chen, C. H. Smith, and S. C. Fralick, "A fast computational algorithm for the discrete cosine transform," *IEEE Trans. Com.*, vol. 25, pp. 1004–1009, Sept. 1977.

[9] W. Chen and C. H. Smith, "Adaptive coding of monochrome and color images," *IEEE Trans. Com.*, vol. 25, pp. 1285–1292, Nov. 1977.

[10] Y. Arai, T. Agui, and M. Nakajima, "A fast DCT-SQ scheme for images," *Trans. IEICE*, vol. E-71, pp. 1095, Nov. 1988.

[11] C. Loeffler, A. Lightenberg, and G. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," *Proc. IEEE ICASSP-89*, vol. 2, pp. 988–991, Feb. 1989.

[12] W. Pennebaker and J. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, New York, 1993.

[13] Z. Wang, "Fast algorithm for the discrete W transform and for the discrete Fourier transform," *IEEE Trans. Accoustics, Speech, and Signal Processing*, vol. 32, pp. 803–816, Aug. 1984.

[14] T. D. Tran, "The BinDCT: fast multiplierless approximation of the DCT," *IEEE Signal Processing Letters*, vol. 7, pp. 141–145, Jun. 2000.

[15] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1993.
[16] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1999.
[17] S. O. Aase and T. A. Ramstad, "On the optimality of nonunitary filter banks in subband coders," *IEEE Trans. on Image Processing*, vol. 4, pp. 1585–1591, Dec. 1995.
[18] T. D. Tran, R. L. de Queiroz, and T. Q. Nguyen, "Linear-phase perfect reconstruction filter bank: lattice structure, design, and application in image coding," *IEEE Trans. on Signal Processing*, vol. 48, pp. 133–147, Jan. 2000.
[19] F. Bruekers and A. Enden, "New networks for perfect inversion and perfect reconstruction," *IEEE Journal on Selected Areas in Communications*, vol. 10, pp. 130–137, Jan. 1992.
[20] K. Komatsu and K. Sezaki, "Reversible discrete cosine transform," *Proc. IEEE ICASSP*, vol. 3, pp. 1769–1772, May. 1998.
[21] K. Komatsu and K. Sezaki, "Design of lossless LOT and its performance evaluation," *Proc. IEEE ICASSP*, vol. 4, pp. 2119–2122, Jun. 2000.
[22] H. S. Malvar, "Biorthogonal and nonuniform lapped transforms for transform coding with reduced blocking and ringing artifacts," *IEEE Trans. on Signal Processing*, vol. 46, pp. 1043–1053, Apr. 1998.
[23] S. C. Chan, T. S. Ng, and C. K. Kwok, "A class of M-channel linear-phase biorthogonal filter banks and their applications to subband coding," *IEEE Trans. on Signal Processing*, vol. 47, pp. 564–571, Feb. 1999.
[24] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans on Circuits Syst. Video Tech.*, vol. 6, pp. 243–250, June 1996.
[25] T. D. Tran and T. Q. Nguyen, "A progressive transmission image coder using linear phase filter banks as block transforms," *IEEE Trans. on Image Processing*, vol. Vol. 8, pp. 1493–1507, Nov. 1999.
[26] H. S. Malvar, "Fast progressive image coding without wavelets," *Proc. of 2000 Data Conpression Conference*, pp. 243–252, Mar. 2000.

TABLE X
CODING RESULTS OF DIFFERENT TRANSFORMS (PSNR IN DB)

| Comp. Ratio | DCT | 9/7 WT | TDLT-III $8\times12$ Cfg. 1 | $8\times12$ Cfg. 3 | $8\times16$ Cfg. 1 | $8\times16$ Cfg. 5 |
|---|---|---|---|---|---|---|
| Lena ($512\times512$) | | | | | | |
| 8 : 1 | 39.61 | **40.41** | 40.01 | 39.80 | 39.89 | 39.72 |
| 16 : 1 | 36.24 | **37.21** | 36.98 | 36.68 | 37.12 | 36.88 |
| 32 : 1 | 32.82 | **34.11** | 33.83 | 33.47 | 34.07 | 33.92 |
| 64 : 1 | 29.59 | **31.10** | 30.73 | 30.30 | 30.98 | 30.88 |
| 128 : 1 | 26.90 | **28.38** | 27.84 | 27.51 | 28.14 | 28.13 |
| Goldhill ($512\times512$) | | | | | | |
| 8 : 1 | 36.24 | 36.55 | **36.62** | 36.42 | 36.60 | 36.43 |
| 16 : 1 | 32.72 | 33.13 | 33.22 | 33.04 | **33.31** | 33.19 |
| 32 : 1 | 30.00 | 30.56 | 30.60 | 30.41 | **30.71** | 30.64 |
| 64 : 1 | 27.85 | 28.48 | 28.44 | 28.24 | **28.57** | 28.54 |
| 128 : 1 | 26.00 | 26.73 | 26.60 | 26.37 | **26.77** | 26.75 |
| Barbara ($512\times512$) | | | | | | |
| 8 : 1 | 36.24 | 36.41 | 37.00 | 36.75 | **37.57** | 37.27 |
| 16 : 1 | 31.14 | 31.40 | 31.91 | 31.74 | **32.91** | 32.67 |
| 32 : 1 | 27.23 | 27.58 | 27.85 | 27.79 | **28.95** | 28.79 |
| 64 : 1 | 24.60 | 24.86 | 25.12 | 25.05 | **25.93** | 25.85 |
| 128 : 1 | 22.71 | 23.35 | 23.26 | 23.09 | **23.74** | 23.67 |
| Cafe ($2560\times2048$) | | | | | | |
| 8 : 1 | 31.43 | 31.74 | 31.81 | 31.57 | **31.75** | 31.53 |
| 16 : 1 | 26.27 | 26.49 | 26.73 | 26.59 | **26.78** | 26.67 |
| 32 : 1 | 22.71 | 23.03 | 23.14 | 23.03 | **23.22** | 23.19 |
| 64 : 1 | 20.24 | 20.67 | 20.63 | 20.55 | **20.73** | 20.72 |
| 128 : 1 | 18.50 | 18.95 | 18.83 | 18.77 | **18.97** | 18.94 |
| Woman ($2560\times2048$) | | | | | | |
| 8 : 1 | 37.93 | **38.28** | 38.14 | 35.47 | 38.03 | 37.77 |
| 16 : 1 | 33.21 | 33.59 | 33.21 | 32.59 | **33.63** | 33.49 |
| 32 : 1 | 29.53 | **29.95** | 29.84 | 29.59 | 29.89 | 29.80 |
| 64 : 1 | 26.92 | **27.33** | 27.13 | 27.19 | 27.22 | 27.21 |
| 128 : 1 | 25.06 | **25.43** | 25.17 | 25.31 | 25.32 | 25.26 |
| First luminance frame of News ($176\times144$) | | | | | | |
| 8 : 1 | 35.94 | 35.94 | **36.54** | 36.22 | 36.27 | 36.13 |
| 16 : 1 | 30.11 | 30.24 | **30.64** | 30.47 | 30.59 | 30.55 |
| 32 : 1 | 25.83 | 26.12 | 26.29 | 26.24 | **26.46** | 26.45 |
| 64 : 1 | 22.99 | 23.03 | 23.49 | 23.40 | **23.52** | 23.48 |
| 128 : 1 | 20.94 | 20.63 | 21.13 | 21.16 | **21.45** | 21.42 |
| First luminance frame of Glasgow ($176\times144$) | | | | | | |
| 8 : 1 | 32.31 | 32.33 | 32.57 | 32.47 | **32.58** | 32.44 |
| 16 : 1 | 27.85 | 27.53 | 28.14 | 28.05 | **28.24** | 28.18 |
| 32 : 1 | 24.94 | 24.82 | 25.34 | 25.18 | **25.53** | 25.41 |
| 64 : 1 | 22.81 | 22.81 | 23.27 | 23.14 | **23.36** | 23.35 |
| 128 : 1 | 21.29 | 21.32 | 21.62 | 21.53 | **21.70** | 21.69 |