Permutation Based Design of Orthogonal Block Transforms and Filter Banks

Article *in* Multidimensional Systems and Signal Processing · January 2001

DOI: 10.1023/A:1008468827787 · Source: CiteSeer

Emin Anarim
Bogazici University

Bulent Sankur
Bogazici University

Multidimensional Systems and Signal Processing, 12, 63–79 (2001)

Permutation Based Design of Orthogonal Block Transforms and Filter Banks

HAKAN ÇAĞLAR, SİNAN GÜNTÜRK, EMİN ANARIM, BÜLENT SANKUR
Boğaziçi University, Department of Electrical and Electronics Engineering, 80815, Bebek, Istanbul-Turkey

Received August 10, 1999; Revised May 2, 2000

Abstract. In this work, a new efficient design technique for orthogonal block transforms, lapped orthogonal transforms and 4-channel perfect reconstruction subband filter banks is developed. The technique consists of permutation and sign change operations on a reference vector. This approach can be thought of as a generalization of the Hadamard transform in the sense that the reference vector $\mathbf{h}_0$ (which will be a prototype low-pass filter also forming one of the basis functions of the transform) will in general have components that are not identically 1's. The design technique, a constructive method based on Hadamard arrays, provides a convenient means to explore new transforms. The merit of our method is that the number of unknowns and equality constraints are both reduced significantly which render the design procedure much more feasible while guaranteeing at the same time linear phase.

Key Words: block transform, lapped transform, subband filter bank, paraunitary, permutation, Hadamard array, coding gain

1. Introduction

Orthogonal multi-channel decompositions, either be in the form of Orthogonal Block Transforms (OBTs), Lapped Orthogonal Transforms (LOTs) or more generally Orthogonal Subband Filter Banks (SFBs), have been the most widely used tools for the analysis and compression of signals and images in the past two decades. The theory of these methods are all well established, with many different strategies and design methods for different applications at hand [1]–[7]. However, the quest for new transforms continues due to several reasons:

- The design and optimization of the filter coefficients in a perfect reconstruction M-channel filter bank is generally a high dimensional problem. Current methods try to overcome this problem by simplifying strategies like cosine modulation of a prototype low pass filter [8]–[11], or factorizations of the associated polyphase matrix [12]–[15]. Design and optimization strategies in the time domain and directly on the impulse responses of the filters are also employed [16]–[18]. See also [19] for a different approach.
- It is always of interest to find computationally efficient transforms, such as filters with integer valued impulse responses or transform implementations with small number of arithmetic operations as well as with other possible VLSI advantages.
- It is desirable to have "matched" and/or adaptive transforms for applications in image and audio compression.

In this paper, we propose and implement novel techniques to design orthogonal block transforms and filter banks that address the points listed above. Our approach is similar to the concept of modulated M-channel paraunitary filter bank designs but it is extended to a "Hadamard-like" modulation via *signed permutation operations*. The underlying theory is based on *Hadamard arrays* [20]. These objects have also been studied under the subject of *real orthogonal designs* in the combinatorics literature [21]. Our method employs a linear phase reference vector $\mathbf{h}_0$ as the first basis function of the transform. The remainder of the transform rows are generated by permutations and sign changes of the first row. A simple example in the case of a 4×4 OBT is given below:

$$\mathbf{A} = \begin{bmatrix} a & b & b & a \\ -b & a & a & -b \\ -a & -b & b & a \\ -b & a & -a & b \end{bmatrix}_{4\times 4},$$

where $\mathbf{h}_0 = [a\ b\ b\ a]$ is the reference vector. Recall that a Hadamard matrix consists of all ± 1's and orthogonality is achieved by means of the sign structure. In this sense, this approach can be interpreted as a generalization of the Hadamard transform (see Section 3).

Using the designs in this paper one can choose, in theory, an arbitrary reference vector $\mathbf{h}_0$ provided it satisfies, together with some size restrictions, the minimal self-orthogonality conditions for belonging to an M-channel paraunitary filter bank (see Sections 4 and 5). In practice, however, we choose this reference vector judiciously, i.e., as the outcome of some optimization procedure, and also with positive symmetry since linear phase is stipulated. Thus, via the choice of a reference vector, desired optimality criteria can be incorporated in the design. The optimality criteria can be, e.g., matching to a statistical process or enhancement of the compression performance. For example, the reference vector can be made to match the statistics of the input process such that the rows of the transform block resemble waveform portions most typically encountered in the sample realizations.

The main contribution of this paper is the development of a new technique for M-channel paraunitary filter bank design, that can be equally well applied to the design of OBTs and LOTs in particular. The novel technique uses Hadamard arrays as an instrument to form a transform or filter bank from a given reference vector. The first advantage is that, while existing techniques are based on intricate optimization search, Hadamard arrays provide a straightforward design technique. (We still optimize, but only over the coefficients of $\mathbf{h}_0$.) Using this technique, new transforms such as multiplier-free transforms with potential applications in lossless coding could also be explored [25]. The second advantage is that this design technique leads to basis functions that use the same coefficient values in different positions, thus allowing for efficient VLSI implementation. Using the theory of Hadamard arrays, the paper also puts some of the ideas in [19] into a more mathematical setting with complete proofs and extensions.

The organization of the paper is as follows: In Section 2, we define Hadamard arrays and fix the notation, and also prove two lemmas to be used later in the designs. Section 3 introduces an OBT design procedure based on signed permutations of a chosen reference vector using Hadamard arrays. The following section extends this idea to the design of $N \times 2N$ LOTs named as Permutation-based Lapped Orthogonal Transforms (P-LOTs). In

Section 5, it is shown that the design of 4-channel Perfect Reconstruction (PR) subband filters can be realized also via Hadamard arrays provided that the filter length is an integer multiple of 4. Section 6 is on the design considerations and results; namely, the optimization procedure and compaction performance results for a first order Markov process signal model. A discussion on the decorrelation of channel outputs to increase compaction performance and the compromise incurred and also advantages of this design technique in the VLSI realization context are also indicated in this section. We finish with the conclusions in Section 7.

2. Hadamard Array Based Unitary Matrices

The mathematical infrastructure of our designs will be based on Hadamard arrays defined below [20]:

Definition. A *Hadamard array* $H[N, k, \lambda]$, based on the indeterminates $\{x[1], x[2], \ldots, x[k], k \leq N\}$ is an $N \times N$ array with entries chosen from $\{\pm x[1], \pm x[2], \ldots, \pm x[k]\}$ in such a way that:

(i) in any row or in any column there are λ repeated entries (with positive or negative signs) for each indeterminate;

(ii) the rows and columns are, respectively, pairwise orthogonal.

In this work, we will concentrate on Hadamard arrays of type $H[N, N, 1]$, that is where N variables are shuffled into N positions, and where each variable appears repeated exactly once in each row and column. An example of a Hadamard array of size $N = 8$ with $\lambda = 1$, based on the indeterminates $\{a, b, c, d, e, f, g, h\}$ is shown below:

$$\mathrm{H}[8,8,1] = \begin{bmatrix} a & b & c & d & e & f & g & h \\ -b & a & d & -c & f & -e & -h & g \\ -c & -d & a & b & g & h & -e & -f \\ -d & c & -b & a & h & -g & f & -e \\ -e & -f & -g & -h & a & b & c & d \\ -f & e & -h & g & -b & a & -d & c \\ -g & h & e & -f & -c & d & a & -b \\ -h & -g & f & e & -d & -c & b & a \end{bmatrix}_{8\times 8}.$$

A Hadamard array $H[N, N, 1]$ can be seen as a mapping of $\mathbf{R}^N$ into an N-dimensional subspace of $N \times N$ square matrices. For an indeterminate vector $\mathbf{x}$, let us denote it by $\mathbf{Q}(\mathbf{x})$, where we have dropped the dependence on N for convenience. Then the elements of the matrix $\mathbf{Q}(\mathbf{x}) = (q_{i,j})_{N\times N} \stackrel{\triangle}{=} (q_i(j))_{N\times N}$ are given as

$$q_i(j) = \alpha_i(j) x[\beta_i(j)], \tag{1}$$

where

$$\alpha_1(j), \alpha_2(j), \ldots, \alpha_N(j) \in \{-1, 1\} \qquad j = 1, 2, \ldots, N \tag{2}$$

are the sign change operations and β_i's denote the permutation functions, $i = 1, 2, \ldots, N$. To illustrate the case, let us enumerate the elements of vector $\mathbf{x}$ as $[1\ 2\ 3\ \ldots\ N]$. Upon the application of the permutation β_i, the elements of the vector $\mathbf{x}$ are re-ordered as $[\beta_i(1)\ \beta_i(2)\ \ldots\ \beta_i(N)]$ which also forms the i'th row of the matrix $\mathbf{Q}(\mathbf{x})$. On top of that, the sign change operations take place. The permutations and sign changes are special in the sense that they have to guarantee pairwise row and column orthogonality. The necessary and sufficient conditions which make the matrix $\mathbf{Q}(\mathbf{x})$ orthonormal are as follows:

LEMMA 1 $\mathbf{Q}(\mathbf{x})$ *is orthonormal if and only if the following conditions are satisfied:*

(i) $\beta_i^{-1}(\beta_j(k)) = \beta_j^{-1}(\beta_i(k)) \qquad 1 \le i, j, k \le N$

(ii) $\alpha_i(k)\alpha_j(k) + \alpha_i(t)\alpha_j(t) = 2\delta_{i-j} \qquad 1 \le i, j, k \le N$ (2)

(iii) $\mathbf{x}\mathbf{x}^T = 1$

where $t = \beta_i^{-1}(\beta_j(k))$. *In these definitions,* β_i^{-1} *stands for the inverse of the permutation function* β_i, *i.e.,* $\beta_i^{-1}(\beta_i) = \beta_i(\beta_i^{-1})$ *is the identity mapping.*

Proof: The elements of $\mathbf{Q}(\mathbf{x})\mathbf{Q}(\mathbf{x})^T$ consist of the inner products of the rows of $\mathbf{Q}(\mathbf{x})$. Denote this matrix by $\mathbf{U}$ and consider an entry $u_{i,j}$ of it. Then by definition,

$$u_{i,j} = \sum_k \alpha_i(k)x[\beta_i(k)]\alpha_j(k)x[\beta_j(k)].$$

Reordering this sum with the permutation $\beta_i^{-1}\beta_j$ and using the symbol t for $\beta_i^{-1}(\beta_j(k))$, one arrives at

$$\begin{aligned} u_{i,j} &= \sum_k \alpha_i(t)x[\beta_i(t)]\alpha_j(t)x[\beta_j(t)] \\ &= \frac{1}{2}\sum_k \{\alpha_i(k)x[\beta_i(k)]\alpha_j(k)x[\beta_j(k)] + \alpha_i(t)x[\beta_i(t)]\alpha_j(t)x[\beta_j(t)]\}. \end{aligned}$$

Finally, using the three conditions above, one gets

$$\begin{aligned} u_{i,j} &= \frac{1}{2}\sum_k \{\alpha_i(k)\alpha_j(k) + \alpha_i(t)\alpha_j(t)\}x[\beta_i(k)]x[\beta_j(k)] \\ &= \delta_{i-j}\sum_k x[\beta_i(k)]x[\beta_j(k)] \\ &= \delta_{i-j}. \end{aligned}$$

This proves that the conditions listed in Lemma 1 are sufficient to make $\mathbf{Q}(\mathbf{x})$ an orthogonal matrix. To prove necessity, consider the fact that the indeterminates are arbitrary and that $q_i(k) = \alpha_i(k)x[\beta_i(k)]$ and $q_j(k) = \alpha_j(k)x[\beta_j(k)]$. The index of the column where $x[\beta_j(k)]$

occurs on the i^{th} row is given by $t = \beta_i^{-1}(\beta_j(k))$. As each index occurs once in each row, we should have $x[\beta_j(t)] = x[\beta_i(k)]$ and $\alpha_i(k)\alpha_j(k) = -\alpha_i(t)\alpha_j(t)$ in order to have the inner product of any two distinct rows identically equal to zero. It is easy to show that the same conditions are valid for all the columns of Hadamard array. This concludes the proof. ■

Remark. Let us define β_i^2 as the composition of β_i with itself. Notice that the item (i) in Lemma 1 implies that β_i^2 is the identity mapping for all i, $1 \le i \le N$. To see this it is enough to note that, without loss of generality, one may take β_1 to be the identity mapping and insert $j = 1$.

It is known that Hadamard arrays of type $H[N, N, 1]$ exist only for $N = 1, 2, 4$ and 8, and that they are equivalent under the basic symmetries, namely:

- Interchanging rows or columns,
- Multiplying rows or columns by -1,
- Replacing any variable by its negative.

The proof is group theoretic and is based on results on quaternion groups and Cayley loops [20].

Using the conditions on the entries of $\mathbf{Q(x)}$ stated in Lemma 1, we prove the following lemma to be used in the development of the proposed LOTs and filter banks.

LEMMA 2 *If* $\mathbf{x}\mathbf{y}^T = 0$, *then*

$$\mathbf{Q}(\mathbf{x})\mathbf{Q}(\mathbf{y})^T + \mathbf{Q}(\mathbf{y})\mathbf{Q}(\mathbf{x})^T = \mathbf{0} \tag{3}$$

Proof: Denote $\mathbf{Q}(\mathbf{x})\mathbf{Q}(\mathbf{y})^T + \mathbf{Q}(\mathbf{y})\mathbf{Q}(\mathbf{x})^T$ by $\mathbf{V} = (v_{i,j})_{N\times N}$ so that

$$v_{i,j} = \sum_k \alpha_i(k)x[\beta_i(k)]\alpha_j(k)y[\beta_j(k)] + \sum_k \alpha_i(k)y[\beta_i(k)]\alpha_j(k)x[\beta_j(k)]. \tag{4}$$

After performing the permutation $\beta_i^{-1}\beta_j$ over the second term of the equation above, one has

$$v_{i,j} = \sum_k \alpha_i(k)x[\beta_i(k)]\alpha_j(k)y[\beta_j(k)] + \sum_k \alpha_i(t)y[\beta_i(t)]\alpha_j(t)x[\beta_j(t)],$$

where again we use the symbol t to denote $\beta_i^{-1}(\beta_j(k))$. Now, it follows by substituting the conditions on α_i's and β_i's that

$$\begin{aligned} v_{i,j} &= \sum_k \{\alpha_i(k)\alpha_j(k) + \alpha_i(t)\alpha_j(t)\}x[\beta_i(k)y[\beta_j(k)] \\ &= 2\delta_{i-j}\sum_k x[\beta_i(k)y[\beta_j(k)] \\ &= 0. \end{aligned}$$

This completes the proof of Lemma 2. ■

Let us note that the statement of Lemma 1 is basic in the theory of Hadamard arrays and also exists in the literature, e.g., [20] in similar forms, though without the explicitly defined permutation operations. On the other hand, Lemma 2 seems original and is essential to the constructions presented in this paper. Using these two lemmas, it will be possible to construct permutation-based OBTs, LOTs and SFBs. In addition, linear phase will be obtained by choosing even symmetric reference vectors. The transform matrix $\mathbf{A}$ will be formed with appropriate Hadamard arrays using a single vector or a single filter unit sample response $\mathbf{h}_0$. In this respect, we first introduce the $N \times N$ block transform.

3. OBT Design Based on Hadamard Arrays

In this particular case of the proposed transforms, called P-OBTs (Permutation-based OBTs), a block transform is constructed by first obtaining a reference vector, which also forms, let us say, the first row of the transform matrix. The other rows are generated by signed permutation operations on the elements of this reference vector. A suitable reference vector $\mathbf{h}_0$ can be found using a constrained optimization technique, e.g. to maximize the energy compaction or to minimize the interband decorrelation efficiency for a given set of input statistics. The only restrictions on the reference vector are its size, which must be an integer power of 2, and its positive symmetry to guarantee the linear phase basis functions of the transform. In a sense, the proposed approach can be thought of as the generalization of the well known Hadamard transform. (Recall that entries of an Hadamard matrix consist only of 1's and −1's.)

Since the reference vector (the first row) was assumed to be symmetric, it can be written as $\mathbf{h}_0 = [\tilde{\mathbf{h}}_0 \quad \tilde{\mathbf{h}}_0\mathbf{J}_{N/2}]$ where $\mathbf{J}_{N/2}$ is the counter identity matrix of dimension $N/2$. It is easily seen from the orthogonality of $\mathbf{Q}(\tilde{\mathbf{h}}_0)$ that the matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q}(\tilde{\mathbf{h}}_0) & \mathbf{Q}(\tilde{\mathbf{h}}_0)\mathbf{J}_{N/2} \\ \mathbf{Q}(\tilde{\mathbf{h}}_0) & -\mathbf{Q}(\tilde{\mathbf{h}}_0)\mathbf{J}_{N/2} \end{bmatrix}_{N\times N} \tag{5}$$

is also orthogonal. Since Hadamard arrays of type $H[N, N, 1]$ exist up to size 8 as mentioned in the previous section, the constructed matrices exist up to size 16. Also, for lower dimensions, an alternative way of attaining linear phase could be to choose the reference vector of an $H[N, N, 1]$ to be symmetric. In a non-constructive approach one would have to search through all $H[N, N/2, 2]$ type of Hadamard arrays that have symmetric rows. Hadamard arrays of size $N \geq 16$ and $\lambda = 2$ remain yet to be explored.

A more general form of the transform matrix can be expressed as:

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q}(\tilde{\mathbf{h}}_0) & \mathbf{Q}(\tilde{\mathbf{h}}_0)\mathbf{J}_{N/2} \\ \mathbf{Q}(\tilde{\mathbf{h}}_1) & -\mathbf{Q}(\tilde{\mathbf{h}}_1)\mathbf{J}_{N/2} \end{bmatrix}_{N\times N} \tag{6}$$

where $\tilde{\mathbf{h}}_1$ is found from $\tilde{\mathbf{h}}_0$ by simply changing the sign of an arbitrary number of elements of $\tilde{\mathbf{h}}_0$. It can be shown that this second form generates $2^{N/2-1}$ different OBT solutions for a given reference vector. This solution set gives us a wide class of possible OBTs based on signed permutation operations on a given reference vector. It turns out that for $N \leq 8$, the shuffling based block transform design technique proposed in [19] is just one of the possible solutions in Eq. (6).

4. LOT Design Based on Hadamard Arrays

The proposed design technique can be applied to the construction of not only OBTs (non-overlapping basis vectors), but also LOTs and SFBs (overlapping basis vectors). Recall that LOT uses basis vectors that overlap in adjacent blocks so that the length of the filters is twice the number of channels, i.e. $N = 2M$. Using ideas similar to those in the previous section, we now construct a Permutation-based Lapped Orthogonal Transform (P-LOT) structure.

Given the reference vector $\mathbf{h}_0$, the remaining $(M-1)$ basis vectors $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_{M-1}$ are embedded in the Hadamard array structure as follows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q}(\mathbf{h}_0') & \mathbf{Q}(\mathbf{h}_0''\mathbf{J})\mathbf{J} & \mathbf{Q}(\mathbf{h}_0''\mathbf{J}) & \mathbf{Q}(\mathbf{h}_0')\mathbf{J} \\ \mathbf{Q}(\mathbf{h}_0') & \mathbf{Q}(\mathbf{h}_0''\mathbf{J})\mathbf{J} & -\mathbf{Q}(\mathbf{h}_0''\mathbf{J}) & -\mathbf{Q}(\mathbf{h}_0')\mathbf{J} \end{bmatrix}_{M\times 2M}, \tag{7}$$

where $\mathbf{Q}$ is an $H[M/2, M/2, 1]$ type Hadamard array and

(i) $\mathbf{h}_0 = [\tilde{\mathbf{h}}_0, \quad \tilde{\mathbf{h}}_0\mathbf{J}_M]$,

(ii) $\tilde{\mathbf{h}}_0 = [\mathbf{h}_0' \quad \mathbf{h}_0'']$; (9)

$\mathbf{h}_0'$ and $\mathbf{h}_0''$ denoting the two equal length partitions of the first half of $\mathbf{h}_0$, and $\mathbf{J}$ the counter identity matrix of appropriate size. Then, $\mathbf{A}$ satisfies the following perfect reconstruction (PR) conditions:

$$\begin{cases} \mathbf{A}\mathbf{A}^T = I_M, \\ \mathbf{A}\mathbf{W}\mathbf{A}^T = \mathbf{0}, \end{cases} \tag{10}$$

provided that

$$\mathbf{h}_0\mathbf{W}^k\mathbf{h}_0^T = \delta_k, \qquad k = 0, 1, \tag{11}$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{0} & \mathbf{I}_M \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{12}$$

is the $2M \times 2M$ *one block shift matrix*.

The proof can be obtained in a straightforward way by evaluating the expressions in Eq. (10) by using Lemma 1 and Lemma 2. Note that the filter length is twice the number of channels. Eq. (10) states that, for an orthogonal PR solution the matrix $\mathbf{A}$ must have basis vectors that are orthogonal for shifts by kM, $k = 0, 1$. In our P-LOT design, the proposed Hadamard array structure implies the orthogonality of the rows of $\mathbf{A}$ ($k = 0$ case), and guarantees the orthogonality with one block shift ($k = 1$ case) when the first row has this property. Finally, since all the M rows of the transform obtained through signed permutations use the same set of multiplier coefficient values, the VLSI implementation of this P-LOT structure can also be realized efficiently (see the discussion in Section 6).

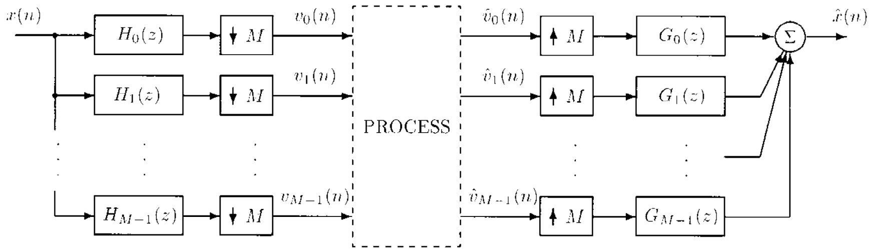

Figure 1. M-channel maximally decimated filter bank structure.

5. 4-Channel Orthogonal SFB Design

In this section, we develop 4-channel orthogonal SFBs with the PR property by using signed permutation operations on a given reference filter in the time domain. Assume a conventional maximally decimated M-band analysis/synthesis system shown in Figure 1. We address the following necessary and sufficient conditions in the time domain for the exact reconstruction of the M-band orthogonal PR-SFB:

$$\sum_m h_i(m)h_j(m - Mk) = \delta_{i-j}\delta_k \quad i, j = 1, 2, \ldots, M, \quad k \in \mathbf{Z}, \tag{13}$$

where $h_i(\cdot)$ is the i^{th} subband filter impulse response. Thus, the filter impulse responses and their translates by integer multiples of M form an orthonormal set. Recall that in the orthogonal case the impulse responses of the synthesis filters are just the time reversed version of the analysis filters.

Now we show that there also exists a 4-channel paraunitary SFB solution with linear phase and PR property using a design technique similar to the ones we have described earlier, but this time also exploiting a special property of $H[4, 4, 1]$, or in other words the *quaternion type Hadamard array*. This technique is again based on the idea of finding a low-pass prototype filter that satisfies self-orthogonality by shifts of $M = 4$, and obtaining the remaining $(M - 1)$ filters by signed permutation operations on the impulse response coefficients. For 4-band designs with $4K$-tap filters, the K orthonormality equations that $\mathbf{h}_0$ needs to satisfy turn out to be sufficient, and no other constraints are necessary. If we had not resorted to this signed permutation design strategy, the number of orthonormality conditions to be satisfied would have been $16K$ (with some symmetries), which is dramatically larger.

Our design procedure for SFBs with $4 \times 4K$ sized transform matrices, where K is an integer, depends on the property that there exists an $H[4, 4, 1]$ of the form

$$\begin{bmatrix} \mathbf{Q}_1(\mathbf{x}) & \mathbf{Q}_1(\mathbf{y}) \\ \mathbf{Q}_2(\mathbf{y}) & -\mathbf{Q}_2(\mathbf{x}) \end{bmatrix}, \tag{14}$$

for which, the following example can be given:

$$\begin{bmatrix} a & b & c & d \\ b & -a & d & -c \\ c & -d & -a & b \\ d & c & -b & -a \end{bmatrix}.$$

Here $\mathbf{Q}_1$ and $\mathbf{Q}_2$ are both Hadamard arrays of order 2. If an $\mathbf{h}_0$ is partitioned as

$$\mathbf{h}_0 = [\mathbf{h}_1^{'}\mathbf{h}_1^{''}\mathbf{h}_2^{'}\mathbf{h}_2^{''} \ldots \mathbf{h}_K^{'}\mathbf{h}_K^{''}]_{1\times 4K} \tag{15}$$

where $\mathbf{h}_i^{'}$ and $\mathbf{h}_i^{''}$ are vectors of size 2, then we must have $\mathbf{h}_i^{'} = \mathbf{h}_{K+1-i}^{''}\mathbf{J}_2$ and $\mathbf{h}_i^{''} = \mathbf{h}_{K+1-i}^{'}\mathbf{J}_2$ to have a positive symmetry. Then we design the $4 \times 4K$ transform matrix to be

$$\mathbf{A} = \begin{bmatrix} \mathbf{Q}_1(\mathbf{h}_1^{'}) & \mathbf{Q}_1(\mathbf{h}_1^{''}\mathbf{J}_2)\mathbf{J}_2 & \ldots\ldots\ldots & \mathbf{Q}_1(\mathbf{h}_K^{'}) & \mathbf{Q}_1(\mathbf{h}_K^{''}\mathbf{J}_2)\mathbf{J}_2 \\ \mathbf{Q}_2(\mathbf{h}_1^{'}) & -\mathbf{Q}_2(\mathbf{h}_1^{''}\mathbf{J}_2)\mathbf{J}_2 & \ldots\ldots\ldots & \mathbf{Q}_2(\mathbf{h}_K^{'}) & -\mathbf{Q}_2(\mathbf{h}_K^{''}\mathbf{J}_2)\mathbf{J}_2 \end{bmatrix}_{4\times 4K}. \tag{16}$$

The PR conditions necessitate that

$$\mathbf{A}\mathbf{W}^k\mathbf{A}^T = \delta_k\mathbf{I}, \quad k = 0, 1, \ldots, K-1, \tag{17}$$

which in turn implies that all necessary PR conditions in Eq. (13), i.e., for the $4 \times 4K$ filter bank design, are reduced to the orthogonality of $\mathbf{h}_0$ by shifts of 4 given as

$$\mathbf{h}_0\mathbf{W}^k\mathbf{h}_0^T = \delta_k, \quad k = 0, 1, \ldots, K-1. \tag{18}$$

Here $\mathbf{W}$ is the $4K \times 4K$ matrix whose right product shifts a vector by 4. The proof is given in the Appendix.

6. Design Considerations and Results

In this section, we will present a number of design examples using the techniques detailed in previous sections and also point out the VLSI implementation advantage of the proposed scheme.

VLSI Implementation Advantage:
The proposed permutation-based design scheme allows for efficient digital implementation in various formats. For unitary matrices based on Hadamard arrays, no decimation is used, hence it is fair to compare it with other block transforms such as the Discrete Cosine Transform (DCT). Consider, for example, an 8×8 transform matrix $\mathbf{A}$ as in Eq. (5). This matrix is composed of 4 distinct entries, hence it suffices to use only 4 fixed multipliers to implement this transform. Note that this feature is useful for serial architectures and is a different type of advantage when compared to the conventional parallel fast algorithms for the computation of DCT that rely on factorizations of the transformation matrix. The proposed transformations also admit partial factorizations that enable some of the computations to be done in parallel. (In [25], we also give some new Hadamard array-based designs

that admit full factorizations.) Thus, mixed type architectures can be employed as well. In [26], an implementation is given for the 8×8 P-OBT and its 2D version, where the data is permuted through a butterfly-like architecture and converted to serial format for subsequent multiplications. The multipliers are fully utilized and work at the original data rate. These multipliers can even be hard-wired, increasing the speed formidably. A comparison was made with a similar architecture for 2D 8×8 DCT implementation. This implementation contains a 1D DCT block with 12 fixed coefficient multipliers and 30 adders and a butterfly block with 4 multipliers and 40 adders, whereas the proposed 2D 8×8 P-OBT implementation contains 8 multipliers and 82 adders. The overall performance was observed to be superior in terms of area/speed as compared to DCT with only a minor loss in performance. On the other hand, other 2D 8×8 DCT implementations referred to in [26] and that are parallel in nature have multiplier counts varying from 96 to 192 and adder counts in the range $462 - 472$. However, it must be noted that the throughput in these architectures is 8 times that of a serial architecture.

For the case of 4-channel orthogonal subband filter design, decimation of the data should naturally be exploited, as detailed in the implementation presented in [27]. Here the data enters the filter with the coefficients being switched to meet the correct data at the correct time, again resulting in 100% utilization of the multipliers. However, the performance of a 1-level 4-band filter was found to be approximately equivalent to a 2-band 3-level dyadic wavelet filter. Reducing the number of multiplications to N (the filter length) is again possible [28], though the consequent routing and switching is rather complicated.

In conclusion the scheme proposed here allows for efficient VLSI implementation for the case of block transforms where the number of multipliers is very small. For the case of M-band filter banks, exploiting the decimation is straightforward without requiring complicated switching and routing. Finally, another advantage is that since the orthogonality does not depend on the word length of the transform coefficients, perfect reconstruction is less sensitive to quantization of the coefficients.

Orthogonal Coder with Decorrelated Outputs:

It is well known that an optimal M-band filter bank will result in uncorrelated outputs. However the converse is not true, in other words uncorrelatedness is not a sufficient condition for optimality, unless one deals with block transforms. Nevertheless it is intuitive to expect that making subband outputs uncorrelated will increase the compaction performance of the coder [24]. If maximization of a coding gain is a more important consideration than efficient implementation, the set of impulse responses obtained through the signed permutation technique can be transformed to remove any correlation between the subband outputs [24]. The decorrelation can be achieved by premultiplying the filter bank matrix by the modal matrix $\mathbf{M}$ of the filter bank output covariance, that is $\mathbf{M}^T\mathbf{A}_0$ becomes the new filter bank matrix, where $\mathbf{A}_0$ is a generic filter bank. The cost of this orthogonalization will, however, be that the filters will no longer share the same numerical coefficient values, which was instrumental for efficient hardware implementation. In the design examples we investigated, we have observed that the coding gain performance of their decorrelated versions was in general better as compared to the performance of the original filters.

The diagonalizing transformation can be interpreted as a better approximation, if not the realization of the globally optimal filter bank from the energy compaction point of view.

Notice that the most energy compacting filter solution $\hat{\mathbf{A}}_0$ would be obtained by an optimizing search in MN dimensional space, where the filter banks of consideration have M channels and N taps. This search effort is not feasible even for small sized transforms. The merit of our method is that the number of unknowns are reduced to $N/2$ and the number of equality constraints to N/M, which renders the optimization search much more feasible.

Optimization Procedure:
The designs of OBTs, LOTs and 4-channel orthogonal SFBs as described in Sections 3, 4 and 5 rely on an optimized reference filter $\mathbf{h}_0$. One method to obtain this prototype response would be via constrained optimization. For a given input covariance matrix $\mathbf{R}_{xx}$, one can maximize the compaction gain under the constraints of normality and orthogonality, i.e.,

$$J = \mathbf{h}_0^T \mathbf{R}_{xx} \mathbf{h}_0 + \lambda_0 (\mathbf{h}_0^T \mathbf{h}_0 - 1) + \sum_{k=1}^{K-1} \lambda_k (\mathbf{h}_0^T \mathbf{W}^k \mathbf{h}_0). \tag{19}$$

Other desirable properties, such as unit step response, uncorrelated subbands can also be imposed as in [23].

Compaction Performance:
A commonly-used performance measure of a subband coder is the coding gain defined as [3], [7],

$$G_{SBC} = \frac{\frac{1}{M}\sum_{k=0}^{M-1} \sigma_k^2}{\left[\prod_{k=0}^{M-1} \sigma_k^2\right]^{1/M}}, \tag{20}$$

where σ_k^2 represents the variance of the output signal of the k-th analysis filter for a particular type of input signal. In the design examples, the block transforms and subband filters have been matched to the input statistics of a Markov-1 source model, i.e., with normalized correlation sequence

$$R(k) = \rho^{|k|}, \qquad k = 0, \pm 1, \ldots,$$

as this model, e.g., with $\rho = 0.95$, is considered to be a general 2nd order representation of image statistics. More specifically, the first row $\mathbf{h}_0$, is derived by using a statistical optimization technique as in [23] to maximize the coding gain.

The coding gain performance of the permutation-based filters/transforms vis-a-vis that of other known transforms such as DCT are shown in Table 1 for AR(1) process. Notice that among the two gain figures in the cells of Table 1, the first one refers to our original design while the second one stands for the gain of the decorrelated filter. This includes the LOT as well; the two figures correspond to the original LOT design (based on the DCT basis functions) and its optimally decorrelated version, as given in [2], [29]. Table 2 depicts the first halves of the symmetric reference vectors $\mathbf{h}_0$ for 4-band designs. The first one corresponds to the 24-tap filter bank design with its magnitude spectra shown in Figure 2. Table 2 also has two of the optimal integer-valued 4-band solutions; see [25] for details and many other solutions. In Table 3 we give, as an example, the transformation matrix defined by Eq. (16) for the integer-valued solution given in the 4th column of Table 2. Notice how the orthogonality conditions are identically satisfied. The main observations are as follows:

- Subband Filters: 4-band 12 to 16 tap filter designs yield superior performance, as expected, compared to DCT and the optimum Karhunen-Loeve Transform (KLT). For example the 4-band 16-tap design provides $G_{SBC} = 6.85$ while those of DCT and KLT remain, respectively, at 5.71 and 5.73. On the other hand the performance of, let us say, 8-tap Smith-Barnwell filters stands at 6.98 [3]. This slight loss of performance of the permutation-based filter against the Smith-Barnwell design is due to the compromise of linear phase property and to the smaller degrees of freedom.

- Decorrelation Advantage: In the “4-BAND” cell the performance figures are the same up to second decimal digit. Consequently, the permutation-based 4-band filters seem to perform very well in the sense that they do not stand to gain from decorrelation operation, up to second decimal digit. Thus they can be used in their original design form that leads to efficient numerical implementation.

- Lapped Orthogonal Transform (second and third rows of the table): For the 4-band case P-LOT is on a par with the LOT. For 8-band and 16-band cases, P-LOT falls short of LOT by about 10% in gain. After decorrelating both schemes (LOT and P-LOT) the performance gap narrows to 5%.

- Orthogonal Block Transforms: P-OBT is definitely inferior to DCT. If one were to decorrelate either of them, one would simply get, of course, the KLT performance. In [25], a different design of block transforms with higher degrees of freedom is advanced. The method again makes use of Hadamard arrays and in terms of coding gain, a difference of 1% is achieved with respect to DCT. ($G_{TC} = 7.57$ for 8×8 design).

- By introducing extra regularity conditions (that is extra zeros at the aliasing frequency locations $\omega = \pi l/2$ $l = 1, 2, 3$) on the prototype low-pass filter $\mathbf{h}_0$, one obtains a 4-band symmetric orthogonal wavelet basis similar to those in [19].

7. Conclusions

In this work, a novel method to construct orthogonal block transforms and PR multichannel filter banks based on permutation operations has been presented. The method is based on obtaining a reference vector, that would typically possess some desired characteristics, and permuting and sign changing its elements to obtain the rest of the filters in the bank. Using this method, sets of $N \times N$ OBTs, $M \times 2M$ LOTs, and 4-channel $(4 \times 4K)$ orthogonal SFBs can be constructed in a direct way, that is without any approximation or tedious optimization schemes. This allows us to explore now avenues such as matched transforms/filters [23], or transforms/filters with integer coefficients [25]. The compromise is that the permutation scheme allows less control over the spectral characteristics of the other (M-1) channels. Block transforms of size $N \leq 16$ and $M \times 2M$ LOT’s of size $M \leq 16$ can be designed directly using, respectively, Eqs. (6), (7), and (8). If transforms/P-LOTs with higher

Table 1. The coding gain performance of the permutation-based filters/transforms vis-a-vis DCT, KLT, LOT and ideal band splitting for AR(1) input source with $\rho = 0.95$. If there are two gain figures in a cell, the first figure relates to the original design, while the second one refers to its decorrelated version.

	4-BAND	8-BAND	16-BAND
8 TAP	6.25-6.25		
12 TAP	6.63-6.63		
16 TAP	6.85-6.85		
20 TAP	6.93-6.93		
24 TAP	7.00-7.00		
32 TAP	7.05-7.05		
P-LOT	5.91-6.22	7.04-8.13	7.55-9.01
LOT	5.89-6.24	7.70-8.39	8.81-9.49
P-OBT	5.22	6.42	6.91
DCT	5.71	7.63	8.82
KLT	5.73	7.66	8.87
IDEAL	7.24	9.16	9.94

Table 2. First halves of the symmetric reference vectors $\mathbf{h}_0$ for various $4 \times 4K$ permutation-based SFB designs and the corresponding coding gain figures: The first one is the 24-tap floating-point arithmetic design (see Table 1) and others are some of the integer-valued solutions given in [25].

n	$h_0(n)$	$h_0(n)$	$h_0(n)$
1	-1.6181744e-02	-2^1	2^1
2	7.7947541e-03	-2^1	-2^0
3	2.2837287e-02	2^0	-2^2
4	1.1000733e-02	2^0	-2^1
5	2.8075635e-02	2^2	-2^0
6	-3.2823687e-02	-2^2	2^3
7	-8.9973315e-02	-2^3	$2^4 + 2^2 + 2^1$
8	-7.0577761e-02	-2^3	$2^4 + 2^3 + 2^1$
9	1.0022727e-02	2^1	
10	1.9518025e-01	$2^4 + 2^1$	
11	3.9974788e-01	$2^5 + 2^3 - 2^0$	
12	5.3488188e-01	$2^5 + 2^4 + 2^0$	
G_{TC}	7.00	6.82	6.77

Table 3. The transformation matrix for the third design example in Table 2. The normalization is given by $\mathbf{AA}^T = 50^2\,\mathbf{I}$.

$$\mathbf{A} = \begin{bmatrix} 2 & -1 & -4 & -2 & -1 & 8 & 22 & 26 & 26 & 22 & 8 & -1 & -2 & -4 & -1 & 2 \\ -1 & -2 & 2 & -4 & 8 & 1 & -26 & 22 & 22 & -26 & 1 & 8 & -4 & 2 & -2 & -1 \\ 2 & 1 & -4 & 2 & -1 & -8 & 22 & -26 & 26 & -22 & 8 & 1 & -2 & 4 & -1 & -2 \\ -1 & 2 & 2 & 4 & 8 & -1 & -26 & -22 & 22 & 26 & 1 & -8 & -4 & -2 & -2 & 1 \end{bmatrix}_{4\times 16}$$

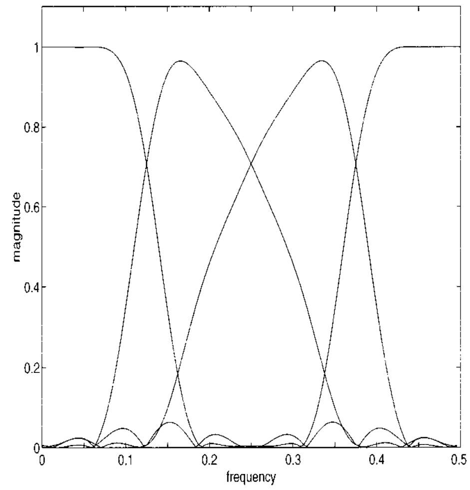

Figure 2. Magnitude spectra of the 4-channel 24-tap filter bank corresponding to the first design example in Table 2.

dimensions are desired, than an extra subset of cross-orthogonality constraints must be satisfied in addition to the self-orthogonality of reference vector $\mathbf{h}_0$ as in Eq. (12) [19]. Based on the design example using the AR(1) model ($\rho = 0.95$) the 4-band filters seem to perform very well, to obviate any gain increase from decorrelation. The performance of the permutation-based LOT and block transforms fall short of their DCT-based counterparts. However we think that the proposed scheme will prove its strength further with integer transforms [25].

Acknowledgements

This work was supported by both TÜBİTAK under contract EEEAG-83, EEEAG-139 and Boğaziçi University Research Foundation under contract 93HA0238. The authors wish to acknowledge both TÜBİTAK and Boğaziçi University Research Foundation for their supports during the progress of this work.

Appendix

Lemma 2 can be generalized as follows:

If

$$\sum_{m,n} \mathbf{x}_m \mathbf{x}_n^T = 0, \tag{A.1}$$

then

$$\sum_{m,n} \mathbf{Q}(\mathbf{x}_m)\mathbf{Q}(\mathbf{x}_n)^T + \mathbf{Q}(\mathbf{x}_n)\mathbf{Q}(\mathbf{x}_m)^T = 0. \tag{A.2}$$

In the lemma, the indices n, m may take completely arbitrary values. The proof is done just in the same way as in Lemma 2, and we have one additional summation. The index change $t = \beta_i^{-1}(\beta_j(k))$ is applied to the second term and one gets the result. Now we prove that

$$\mathbf{A}\mathbf{W}^k\mathbf{A}^T = \delta_k \mathbf{I}, \quad k = 0, 1, \ldots, K-1, \tag{A.3}$$

given that

$$\mathbf{h}_0\mathbf{W}^k\mathbf{h}_0^T = \delta_k, \tag{A.4}$$

where $\mathbf{A}$ is given in Eq. (16).

If we apply the partitioning of $\mathbf{h}_0$ as in Eq. (15) to $\mathbf{h}_0\mathbf{W}^k\mathbf{h}_0^T = \delta_k$, we get

$$\sum_{1 \le i \le K-k} \mathbf{h}'_i\mathbf{h}'^T_{i+k} + \mathbf{h}''_i\mathbf{h}''^T_{i+k} = \delta_k. \tag{A.5}$$

Equivalently this may be written as

$$\sum_{1 \le i \le K-k} \mathbf{h}'_i\mathbf{h}'^T_{i+k} + \sum_{1 \le i \le K-k} \mathbf{h}''_{K+1-k-i}\mathbf{h}''^T_{K+1-i} = \delta_k,$$

where we have changed the order of summation in the second sum and replaced i with $K+1-k-i$.

Now $\mathbf{h}''_{K+1-i} = \mathbf{h}'_i\mathbf{J}_2$, $\mathbf{h}''_{K+1-i-k} = \mathbf{h}'_{i+k}\mathbf{J}_2$ and exploiting the fact that $\mathbf{h}'_i\mathbf{h}'^T_j = \mathbf{h}'_j\mathbf{h}'^T_i$, we get

$$\sum_{1 \le i \le K-k} 2\mathbf{h}'_i\mathbf{h}'^T_{i+k} = \delta_k. \tag{A.6}$$

$\mathbf{A}$ being partitioned as given, we look at $\mathbf{A}\mathbf{W}^k\mathbf{A}^T$. If we again divide it into 2×2 matrices as

$$\mathbf{A}\mathbf{W}^k\mathbf{A}^T = \begin{bmatrix} \mathbf{U}^{(k)}_{1,1} & \mathbf{U}^{(k)}_{1,2} \\ \mathbf{U}^{(k)}_{2,1} & \mathbf{U}^{(k)}_{2,2} \end{bmatrix}_{4\times4} \tag{A.7}$$

then

$$\begin{aligned}
\mathbf{U}_{l,l}^{(k)} &= \sum_{1\le i\le K-k} \mathbf{Q}_l(\mathbf{h}_i')\mathbf{Q}_l(\mathbf{h}_{i+k}')^T + \mathbf{Q}_l(\mathbf{h}_i''\mathbf{J})\mathbf{Q}_l(\mathbf{h}_{i+k}''\mathbf{J})^T \\
&= \sum_{1\le i\le K-k} \mathbf{Q}_l(\mathbf{h}_i')\mathbf{Q}_l(\mathbf{h}_{i+k}')^T + \sum_{1\le i\le K-k} \mathbf{Q}_l(\mathbf{h}_{K+1-k-i}''\mathbf{J})\mathbf{Q}_l(\mathbf{h}_{K+1-i}''\mathbf{J})^T \\
&= \sum_{1\le i\le K-k} \mathbf{Q}_l(\mathbf{h}_i')\mathbf{Q}_l(\mathbf{h}_{i+k}')^T + \mathbf{Q}_l(\mathbf{h}_{i+k}')\mathbf{Q}_l(\mathbf{h}_i')^T, \qquad \text{(A.8)}
\end{aligned}$$

which is $\delta_k \mathbf{I}_2$ from the generalized lemma, Eq.'s (A.1), (A.2).

Also for $\mathbf{U}_{l,l'}^{(k)}$, where $l \neq l'$, we write

$$\mathbf{U}_{l,l'}^{(k)} = \sum_{1\le i\le K-k} \mathbf{Q}_l(\mathbf{h}_i')\mathbf{Q}_{l'}(\mathbf{h}_{i+k}')^T - \mathbf{Q}_l(\mathbf{h}_i''\mathbf{J})\mathbf{Q}_{l'}(\mathbf{h}_{i+k}''\mathbf{J})^T.$$

The same steps of index changes and substitutions result in

$$\mathbf{U}_{l,l'}^{(k)} = \sum_{1\le i\le K-k} \mathbf{Q}_l(\mathbf{h}_i')\mathbf{Q}_{l'}(\mathbf{h}_{i+k}')^T - \mathbf{Q}_l(\mathbf{h}_{i+k}')\mathbf{Q}_{l'}(\mathbf{h}_i')^T. \qquad \text{(A.9)}$$

Finally, since there exists an $H[4, 4, 1]$ of the form stated, we easily see that

$$\begin{aligned}
\mathbf{Q}_1(\mathbf{x})\mathbf{Q}_2(\mathbf{y})^T - \mathbf{Q}_1(\mathbf{y})\mathbf{Q}_2(\mathbf{x})^T &= 0 \\
\mathbf{Q}_2(\mathbf{y})\mathbf{Q}_1(\mathbf{x})^T - \mathbf{Q}_2(\mathbf{x})\mathbf{Q}_1(\mathbf{y})^T &= 0, \qquad \text{(A.10)}
\end{aligned}$$

which results in $\mathbf{U}_{l,l'}^{(k)} = 0$ for all k. Hence $\mathbf{A}\mathbf{W}^k\mathbf{A}^T = \delta_k \mathbf{I}_4$.

References

1. P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall, 1992.
2. H. S. Malvar, *Signal Processing with Lapped Transforms*, Butler, WI: Artech House, 1992.
3. A. N. Akansu and R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands and Wavelets*, Springer, 1992.
4. G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*, Wellesley-Cambridge Press, 1996.
5. K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications*, San Diego, CA: Academic Press, 1990.
6. J. W. Woods, *Subband Image Coding*, Boston, MA: Kluwer Academic Publishers, 1991.
7. M. Vetterli and J. Kovačević, *Wavelets and Subband Coding*, Prentice Hall, 1995.
8. R. D. Koilpillai and P. P. Vaidyanathan, "Cosine Modulated FIR Filter Banks Satisfying Perfect Reconstruction," *IEEE Trans. on Signal Proc.*, vol. 40, no. 4, April 1992, pp. 770–783.
9. T. A. Ramstad, "Cosine Modulated Analysis-Synthesis Filter Banks with Critical Sampling and Perfect Reconstruction," *Proc. ICASSP*, Toronto, Canada, May 1991, pp. 1789–1792.
10. Y. P. Lin and P. P. Vaidyanathan, "Linear Phase Cosine Modulated Maximally Decimated Filter Banks with Perfect Reconstruction," *IEEE Trans. on Signal Proc.*, vol. 42, no. 11, Nov. 1995, pp. 2525–2539.
11. T. Q. Nguyen and R. D. Koilpillai, "The Theory and Design of Arbitrary-Length Cosine-Modulated Filter Banks and Wavelets, Satisfying Perfect Reconstruction," *IEEE Trans. on Signal Proc.*, vol. 44, no. 3, Mar. 1996, pp. 473–483.

12. A. K. Soman, P. P. Vaidyanathan, and T. Q. Nguyen, “Linear-Phase Paraunitary Filter Banks: Theory, Factorizations and Applications,” *IEEE Trans. on Signal Proc.*, vol. 41, no. 12, Dec. 1993, pp. 3480–3496.
13. R. L. de Queiroz, T. Q. Nguyen, and K. R. Rao, “The GenLOT: Generalized Linear-Phase Lapped Orthogonal Transform,” *IEEE Trans. on Signal Proc.*, vol. 44, no. 3, Mar. 1996, pp. 497–507.
14. L. Chen, K.-P. Chan, and T. Q. Nguyen, “A Generalized Algorithm for Linear-Phase Paraunitary Filter Banks,” *IEEE Trans. on Signal Proc.*, vol. 46, no. 4, Apr. 1998, pp. 1154–1158.
15. T. D. Tran, R. L. de Queiroz, and T. Q. Nguyen, “The Generalized Lapped Biorthogonal Transform,” *Proc. ICASSP*, Seattle, May 1998.
16. K. Nayebi, T. Barnwell, and M. J. Smith, “Time-domain Filter Bank Analysis: A New Design Theory,” *IEEE Trans. on Signal Proc.*, vol. 40, no. 6, June 1992, pp. 1412–1429.
17. P. Saghizadeh and A. N. Willson, Jr., “A New Approach to the Design of Critically Sampled M-Channel Uniform-Band Perfect-Reconstruction Linear-Phase FIR Filter Banks,” *IEEE Trans. on Signal Proc.*, vol. 46, no. 6, June 1998, pp. 1544–1557.
18. T. Q. Nguyen, “Digital Filter Banks Design - Quadratic-Constrained Formulation,” *IEEE Trans. on Signal Proc.*, vol. 43, no. 9, Sept. 1995, pp. 2103–2108.
19. O. Alkın and H. Çaglar, “Design of Efficient M-Band Coders with Linear Phase and Perfect Reconstruction Properties,” *IEEE Trans. on Signal Proc.*, vol. 43, July 1995, pp. 1579–1590.
20. W. D. Wallis, A. P. Street, and J. S. Wallis, *Combinatorics, Room Squares, Sum Free Sets, Hadamard Matrices*, Springer Verlag, 1972.
21. C. J. Colbourn and J. H. Dinitz, eds., *The CRC Handbook of Combinatorial Designs*, CRC Press, 1996.
22. S. Güntürk, H. Çağlar, E. Anarım, and B. Sankur, “A New Permutational Based Block Transform Design Technique,” *European Conference on Circuit Theory and Design, ECCTD ’95*, Istanbul, 1995, pp. 825–828.
23. H. Çaglar, Y. Liu, and A. N. Akansu, “Statistically Optimized PR-QMF Design,” *Proc. SPIE Visual Comm. and Image Proc.*, 1991, pp. 86–94.
24. M. Unser, “An Extension of the Karhunen-Loeve Transform for Wavelets and Perfect Reconstruction Filter Banks,” *SPIE Vol. 2034*, Mathematical Imaging, 1993, pp. 45–56.
25. S. Güntürk, H. Çağlar, E. Anarım, and B. Sankur, “Design of Multiplier-free Orthogonal Block Transforms and Filter Banks,” Technical Report, BUSİM-96-1, Boğaziçi University, Istanbul, Turkey, 1996. url: http://www.busim.ee.boun.edu.tr/publications/sankur/integer_transforms/
26. E. Yazicioğlu, S. Balkır, G. Dündar, and H. Çağlar, “Implementation of a New Orthogonal Shuffled Block Tranform for Image Coding Applications,” *Journal of Real Time Imaging*, vol. 6, Feb. 2000, pp. 39–46.
27. Y. Atabek, G. Dündar, S. Balkır, H. Çağlar, and E. Anarım, “Design of M-Band Wavelet Filter with Perfect Reconstruction Architecture,” *Proc. International Conference on Telecommunications (ICT ’96)*, Istanbul, April 1996, pp. 225–228.
28. K. Parkhi and T. Nishitani, “VLSI Architectures for Discrete Wavelet Transforms,” *IEEE Trans. on VLSI Systems*, vol. 1, June 1993, pp. 191–202.
29. H. S. Malvar and D. H. Staelin, “The LOT: Transform Coding without Blocking Effects,” *IEEE Trans. on Acoustics, Speech and Signal Proc.*, vol. 37, no. 4, April 1989, pp. 553–559.