A Study on Circular Filtering in Orthogonal Transform Domain

Bong Seop Song and Sang Uk Lee

Abstract

In this paper, we dicuss on the properties related to the circular filtering in orthogonal transform domain. The efficient filtering schemes in six orthogonal transform domains are presented by generalizing the convolution-multiplication property of the DFT. In brief, the circular filtering can be accomplished by multiplying the transform domain filtering matrix W, which is shown to be very sparse, yielding the computational gains compared with the time domain processing. As an application, decimation and interpolation techniques in orthogonal transform domains are also investigated.

I. Introduction

Recently, orthogonal transforms have received considerable interests, due to their various applications in digital signal processing areas.

For example, it is well known that the digital filtering can often be implemented more efficiently by an indirect computational procedure in orthogonal transform domain, which is called the generalized linear filtering[6].

The generalized linear filtering is a technique which computes linear operations indirectly utilizing orthogonal transforms, instead of direct computation in time domain. Figure 1 shows a block diagram of the technique. In the generalized linear filtering, the $N\times 1$ input vector $\underline{X}$ undergoes an orthogonal transformation, resulting in a vector of $N\times 1$ transform coefficients $\underline{X}$. Then, the coefficients are multiplied by an $N\times N$ transform domain filtering matrix W, and an inverse orthogonal transformation is performed to obtain the $N\times 1$ output vector $\underline{y}$. Note that the savings in computation is obtained in the second step. The efficiency of the generalized linear filtering depends on the sparseness of the transform domain filtering matrix W. In other words, the matrix W should have many zero elements for the generalized linear filtering technique to be effective. In this paper, based on the generalized linear filtering, we shall investigate the properties related to the circular filtering in orthogonal transform domain.

Manuscript received May 17, 1996; accepted June 24, 1996.

The authors are the Department of Control and Instrumentation Engineering, School of Electrical Engineering, Seoul National University, Seoul, Korea.

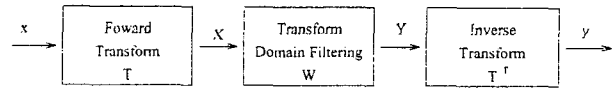

Fig. 1. The block diagram of generalized linear filtering.

It is well known that the discrete Fourier transform(DFT) possesses a convolution-multiplication property. More specifically, a circular convolution of two finite sequences can be implemented efficiently by means of a transform domain multiplication, since the matrix W for the DFT is diagonal. Unfortunately, this property does not hold for other orthogonal transforms, such as the DCT and the DST(discrete sine transform), and the circular filtering technique in general orthogonal transform domain has not been reported yet.

There are several works relating to the filtering in the DCT domain[1,2,3], but these works failed to provide an exact circular filtering in the DCT domain, since the approaches are based on the approximations. In [4], global structures to implement the linear filtering in transform domains were proposed. Recently, symmetric filtering techniques in the DCT and the DST domain were also reported[5]. However, to our best knowledge, a technique for exact circular filtering in orthogonal transform domain has not been shown yet.

In generalized linear filtering technique, the elements of the matrix W are functions of the basis of the orthogonal transform and the filter coefficients. Thus, the approach in this paper is to derive the elements of the matrix W for each orthogonal transform. We consider six orthogonal transforms: DCT, DST, Hartley transform, Hadamard transform, Haar transform, and Slant transform. Since the computational

efficiency of the generalized linear filtering depends on the sparseness of the matrix W, we specifically focus our effort to identify the zero elements in the matrix W.

It will be shown that, in fact, there exist many zero elements in the matrix W, indicating that the generalized linear filtering technique can often be implemented more efficiently than the direct computation in time domain. To say briefly, the highest efficiency is achieved in the Hartley transform domain, since the matrix W is a diagonal matrix. Also, for the DCT and Hadamard transform, the matrix W's are observed to be very sparse, compared to other transforms, yielding the computational gains.

This paper is composed of three sections. First, we describe a transform domain filtering matrix W for a circular filtering in orthogonal transform domain in section II. Secondly, the circular filtering techniques in six orthogonal transforms is presented, together with the comparison of their complexity, in section III. To demonstrate the application, decimation and interpolation techniques in orthogonal transform domains is discussed in section IV, and the conclusions are drawn in section V.

II. Transform Domain Filtering Matrix

The generalized linear filtering concerned in this paper is a circular filtering in orthogonal transform domain. The orthogonal transform is obtained by multiplying $N\times N$ transform matrix T, which is composed of orthonormal basis vectors t_i's in each column. The circular filtering can also be represented in terms of vector-space notation, given by

$$\underline{y}=H\underline{x} \tag{1}$$

where x is the input vector of length N, H is the $N\times N$ circular filtering matrix, and y is the filtered output of the same length as $\underline{x}$. The circular filtering matrix H is given by

$$H=\begin{pmatrix} h_0 & h_{-1} & \cdots & h_{-K} & 0 & \cdots & 0 & h_M & h_{M-1} & \cdots & h_1 \\ h_1 & h_0 & h_{-1} & \cdots & h_{(-K+1)} & h_{-K} & 0 & \cdots & 0 & h_M & \cdots & h_2 \\ \vdots & & & & & & & & & & \vdots \\ h_{-1} & & \cdots & h_{-K} & 0 & & \cdots & 0 & h_M & & & h_0 \end{pmatrix}, \tag{2}$$

where h_k's are the filter coefficients.

It is not easy to deal with the matrix H in terms of elements. For the sake of convenience, we define a useful circular shift vector operator. For a given vector f of length N, let $\tilde{f}^k$ be the vector whose elements are kth circular shifts of vector f's as follows:

$$\tilde{f}_{l'}=\begin{cases} f_{(l-k+N)}, & \text{if } l<k \\ f_{(l-k)}, & \text{if } l\geq k \end{cases} \tag{3}$$

Then, we can easily express the matrix H as

$$H=[\,\underline{h}\ \ \tilde{\underline{h}}^1\ \ \tilde{\underline{h}}^2\cdots\ \tilde{\underline{h}}^{N-1}] \tag{4}$$

where $\underline{h}=[h_0\cdots h_M\,0\cdots 0\,h_{-K}\cdots h_{-1}]$.

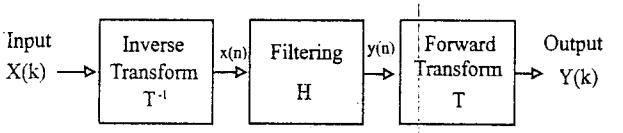

Fig. 2. Circular filtering in orthogonal transform domain.

By denoting the transformed input and output by $N\times 1$ vector $\underline{X}$ and $\underline{Y}$, respectively, Figure 2 shows the block diagram for circular filtering in general orthogonal transform domain[4], which consists of three stages : inverse transform, filtering in time domain, and forward transform. Thus, the relation between the input vector $\underline{X}$ and the output vector $\underline{Y}$ is given by

$$\underline{Y}=W\underline{X}, \tag{5}$$

where

$$W=THT^T. \tag{6}$$

It is seen that the transform domain filtering matrix W completely specifies the filtering in each orthogonal transform domain. Thus, to find the elements w_{ij} of the matrix W, let us begin with the following relations, given by

$$\begin{aligned} W=&\ [\,\underline{t}_0\cdots\underline{t}_{N-1}\,]^T[\,\underline{h}\cdots\tilde{\underline{h}}^{N-1}\,] \\ &\cdot[\,\underline{t}_0\cdots\underline{t}_{N-1}\,] \end{aligned} \tag{7}$$

Then,

$$\begin{aligned} w_{ij}&=[\,\underline{t}_i\cdot\underline{h}\ \underline{t}_i\cdot\tilde{\underline{h}}^1\cdots\underline{t}_i\cdot\tilde{\underline{h}}^{N-1}\,]\cdot\underline{t}_j \\ &=[\,\underline{t}_i\cdot\underline{h}\ \tilde{\underline{t}}_i{}^{N-1}\cdot\underline{h}\cdots\tilde{\underline{t}}_i{}^1\cdot\underline{h}\,]\cdot\underline{t}_j \\ &=(t_{j0}\,\underline{t}_i+t_{j1}\,\tilde{\underline{t}}_i{}^{N-1}+\cdots+t_{j(N-1)}\,\tilde{\underline{t}}_i{}^1)\cdot\underline{h} \end{aligned} \tag{8}$$

but, it is equivalent to

$$\begin{aligned} w_{ij}&=\sum_{k=-K}^{M}h(k)(t_{jk}t_{ik}+t_{j1}t_{i(k+1)}+\text{ldots}+t_{j(n-1)}t_{i(k-1)}) \\ &=\sum_{k=-K}^{M}h(k)a_k(i,j) \end{aligned} \tag{9}$$

where $a_{k(i,j)}=(\,\underline{t}_i\cdot\tilde{\underline{t}}_{j^*})$

Thus, the elements of the matrix W are represented by linear combination of filter sequences, while the combination coefficients $a_k(i,j)$'s are expressed only in terms of the transform bases. Thus, the circular filtering in each transform domain can be characterized by examining the $a_k(i,j)$'s .

III. Circular Filtering in Orthogonal Transform Domain with Linear Phase FIR Filters

Since the filters with linear phase are desirable for image processing[6], in this paper, we consider 4 different types of linear phase FIR filters, namely OS, ES, OA, and EA[7], which are described in Table 1.

Table 1. Linear phase FIR filters.

type	length	symmetry
1(OS)	$L = 2K + 1(M = K)$	$h(k) = h(-k)$
2(ES)	$L = 2K(M = K - 1)$	$h(k) = h(-k - 1)$
3(OA)	$L = 2K + 1(M = K)$	$h(k) = -h(-k)$
4(EA)	$L = 2K(M = K - 1)$	$h(k) = -h(-k - 1)$

From the properties of linear phase filters, we can easily obtain the w_{ij}'s for 4 different types as shown in Table 2. For an example, in the case of the type 1, it is easy to show that

$$\begin{aligned} w_{ij} &= \sum_{k=-K}^{M} h(k)(\underline{t}_i \cdot \tilde{\underline{t}}_{j^*}) \\ &= h(0)(\underline{t}_i \cdot \underline{t}_j) + \sum_{k=1}^{K} h(k)(\underline{t}_i \cdot \tilde{\underline{t}}_{j^*} + \underline{t}_i \cdot \tilde{\underline{t}}_{j^*}) \quad (10) \\ &= h(0)(\underline{t}_i \cdot \underline{t}_j) + \sum_{k=1}^{K} h(k)(\underline{t}_i \cdot \tilde{\underline{t}}_{j^*} + \tilde{\underline{t}}_{i^*} \cdot \underline{t}_j). \end{aligned}$$

Thus, we have

$$w_{ij} = \sum_{k=0}^{K} h(k) a_k(i,j), \quad (11)$$

where $a_k(i,j) = \begin{cases} \underline{t}_i \cdot \underline{t}_j = \delta_{ij}, & k=0 \\ \underline{t}_i \cdot \tilde{\underline{t}}_j^{\,k} + \tilde{\underline{t}}_i^{\,k} \cdot \underline{t}_j, & k \neq 0 \end{cases}$

Table 2. The elements of W with linear phase filters.

type	w_{ij}	$a_k(i,j)$
1(OS)	$\sum_{k=0}^{K} h(k)a_k(i,j)$	$\begin{cases} \underline{t}_i \cdot \underline{t}_j = \delta_{ij} & , k = 0 \\ \underline{t}_i \cdot \tilde{\underline{t}}_j^{k} + \tilde{\underline{t}}_i^{k} \cdot \underline{t}_j & , k \neq 0 \end{cases}$
2(ES)	$\sum_{k=0}^{K-1} h(k)a_k(i,j)$	$\underline{t}_i \cdot \tilde{\underline{t}}_j^{k} + \tilde{\underline{t}}_i^{k+1} \cdot \underline{t}_j$
3(OA)	$\sum_{k=1}^{K} h(k)a_k(i,j)$	$\underline{t}_i \cdot \tilde{\underline{t}}_j^{k} - \tilde{\underline{t}}_i^{k} \cdot \underline{t}_j$
4(EA)	$\sum_{k=0}^{K-1} h(k)a_k(i,j)$	$\underline{t}_i \cdot \tilde{\underline{t}}_j^{k} - \tilde{\underline{t}}_i^{k+1} \cdot \underline{t}_j$

It is seen that the linear combination coefficients are obtained from the transform bases t_i's. Since we deal with six different orthogonal transforms, the basis vectors for each orthogonal transform are listed in Table 3.

Table 3. The basis vectors of orthogonal transforms.

Transform	t_{il}
DCT	$\sqrt{\frac{2}{N}} k_i \cos(\frac{(2l+1)i\pi}{2N})$ $k_0 = \sqrt{\frac{1}{2}}$, $k_i = 1 (i \neq 0)$
DST	$\sqrt{\frac{2}{N+1}} \sin(\frac{(i+1)(l+1)\pi}{N+1})$
Hartley Transform	$\sqrt{\frac{1}{N}}(\cos(\frac{2\pi il}{N}) + \sin(\frac{2\pi il}{N}))$
Hadamard Transform Haar Transform Haar Transform	recursive

Now, let us investigate the properties relating to the circular filtering in each transform domain in more detail.

Before we attempt to derive the linear combination coefficient $a_k(i,j)$, it should be noted that if for a specific i^* and j^*, $a_k(i^*,j^*)$'s are zeros for all k, the element $w_{i^*j^*}$ of the matrix W is always zero, regardless of $h(k)$ from (11). In this way, we can identity the zero elements of the matrix W easily.

1. DCT domain

Let us examine the coefficient $a_k(i,j)$ for the type 1 filter in the DCT domain first. We now start from the relation given by

$$\begin{aligned} a_k(i,j) &= \underline{t}_i \cdot \tilde{\underline{t}}_{j^*} + \tilde{\underline{t}}_{i^*} \cdot \underline{t}_j \\ &= \frac{2}{N} k_i k_j \Big(\sum_{l=0}^{k-1} \cos(\frac{(2l+1)i\pi}{2N}) \cos(\frac{(2(N-k+l)+1)j\pi}{2N}) \\ &\quad + \sum_{l=k}^{N-1} \cos(\frac{(2l+1)i\pi}{2N}) \cos(\frac{(2(l-k)+1)j\pi}{2N}) \\ &\quad + \sum_{l=0}^{k-1} \cos(\frac{(2(N-k+l)+1)i\pi}{2N}) \cos(\frac{(2l+1)j\pi}{2N}) \\ &\quad + \sum_{l=k}^{N-1} \cos(\frac{(2(l-k)+1)i\pi}{2N}) \cos(\frac{(2l+1)j\pi}{2N}) \end{aligned} \quad (12)$$

Using the following relations:

$$\begin{aligned} \sum_{l=S}^{E} \sin(al+b) &= \frac{2}{1-\cos a} \sin(\frac{a}{2}) \sin(\frac{aL}{2}) \sin(\frac{a(S+E)}{2} + b) \\ \sum_{l=S}^{E} \cos(al+b) &= \frac{2}{1-\cos a} \sin(\frac{a}{2}) \sin(\frac{aL}{2}) \cos(\frac{a(S+E)}{2} + b) \end{aligned} \quad (13)$$

where $L = E\text{-}S+1$, we can simplify (12).

If i equals to j, it is easy to show that

$$a_k(i,i)=\begin{cases} N\cos\left(\frac{ki\pi}{N}\right), & i=even \\ (N-2k)\cos\left(\frac{ki\pi}{N}\right), & i=odd. \end{cases} \quad (14)$$

Otherwise, with some algebraic manipulations, $a_k(i,j)$ is expressed as

$$a_k(i,j) = A(k,i,j)\cos\left(\frac{(i+j)\pi}{2}\right)+B(k,i,j)\cos\left(\frac{(i-j)\pi}{2}\right) \quad (15)$$

$$= C(k,i,j)\sin\left(\frac{i\pi}{2}\right)+D(k,i,j)\sin\left(\frac{j\pi}{2}\right), \quad (16)$$

where

$$A(k,i,j) = \frac{2}{1-\cos\left(\frac{(i+j)\pi}{N}\right)}\sin\left(\frac{(i+j)\pi}{2N}\right)\left\{\sin\left(\frac{k(i+j)\pi}{2N}\right)\cos\left(\frac{(N-k)(i-j)\pi}{2N}\right)+\cos\left(\frac{k(i-j)\pi}{2N}\right)\sin\left(\frac{(N-k)(i+j)\pi}{2N}\right)\right\},$$

$$B(k,i,j) = \frac{2}{1-\cos\left(\frac{(i-j)\pi}{N}\right)}\sin\left(\frac{(i-j)\pi}{2N}\right)\left\{\sin\left(\frac{k(i-j)\pi}{2N}\right)\cos\left(\frac{(N-k)(i+j)\pi}{2N}\right)+\cos\left(\frac{k(i+j)\pi}{2N}\right)\sin\left(\frac{(N-k)(i-j)\pi}{2N}\right)\right\}$$

$$C(k,i,j) = \frac{2}{1-\cos\left(\frac{(i+j)\pi}{N}\right)}\sin\left(\frac{(i+j)\pi}{2N}\right)\cos\left(\frac{(i+j)\pi}{2}\right)\cos\left(\frac{(N-2k)i\pi}{2N}\right)+\frac{2}{1-\cos\left(\frac{(i-j)\pi}{N}\right)}\sin\left(\frac{(i-j)\pi}{2N}\right)\cos\left(\frac{(i-j)\pi}{2}\right)\cos\left(\frac{(N-2k)i\pi}{2N}\right),$$

$$D(k,i,j) = \frac{2}{1-\cos\left(\frac{(i+j)\pi}{N}\right)}\sin\left(\frac{(i+j)\pi}{2N}\right)\cos\left(\frac{(i+j)\pi}{2}\right)\cos\left(\frac{(N-2k)i\pi}{2N}\right)-\frac{2}{1-\cos\left(\frac{(i-j)\pi}{N}\right)}\sin\left(\frac{(i-j)\pi}{2N}\right)\cos\left(\frac{(i-j)\pi}{2}\right)\cos\left(\frac{(N-2k)i\pi}{2N}\right)$$

From the properties of sine and cosine functions, (15) and (16) imply

$$w_{ij}=0 \quad , \quad \text{if } i=even \text{ or } j=even, \text{ and } i\neq j. \quad (17)$$

Thus, the circular filtering with the type 1 filters in the DCT domain can be carried out, according to the relation

$$Y(m)=\begin{cases} w_{mm}X(m) & , m=even \\ \sum_{j=0}^{\frac{N}{2}-1} w_{m(2j+1)}X(2j+1) & , m=odd. \end{cases} \quad (18)$$

Similarly, the matrix W for other three filter types are easily found, which are presented in Table 4. The results show that there exist many zero elements in the matrix W for each type of filter, making the matrix W sparse. Thus, the circular filtering in the DCT domain provides some computational gain over the direct time domain processing. Especially, in the case of type 1 filter, it is seen that about $\frac{3}{4}$ of the elements in the matrix W are zeros and in the case of type 2 filter, about $\frac{1}{2}$ of the elements are zeros. However, type 2 and type 4 filter yield less zero elements.

Table. 4. Circular filtering in the DCT domain. E(even number), D(odd number)

t	zero elements	$Y(m)$
1	$i=E$, $i\neq j$	$w_{mm}X(m)$, E
	$j=E$, $i\neq j$	$\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j+1)}X(2j+1)$, D
2	$i=0$	$w_{00}X(0)$, 0
	$j=0$	$\sum_{j=1}^{N-1} w_{mj}X(j)$, D
	$i,j=D$, $i\neq j$	$w_{mm}X(m)+\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j+1)}X(2j+1)$, E
3	$i=E$, $j=E$	0 , 0
	$i=D$, $j=D$	$\sum_{j=1}^{\frac{N}{2}-1} w_{m(2j)}X(2j)$, D
	$i=0$	$\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j+1)}X(2j+1)$, E.
	$j=0$	
4	$i=0$	0 , 0
	$j=0$	$\sum_{j=1}^{N-1} w_{mj}X(j)$, D
	$i,j=E$, $i\neq j$.	$w_{mm}X(m)+\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j+1)}X(2j+1)$, D.

2. *DST domain*

In the DST domain, we use the same approach as in the DCT domain. For an example, with the type 1 filters, the $a_k(i,j)$ is represented as

$$\begin{aligned} a_k(i,j) &= t_i\cdot\tilde{t}_j+\tilde{t}_i\cdot t_j \\ &= \frac{2}{N+1}\Big(\sum_{l=0}^{k-1}\sin\left(\frac{(l+1)(i+1)\pi}{N+1}\right)\sin\left(\frac{(N-k+l+1)(j+1)\pi}{N+1}\right) \\ &+\sum_{l=k}^{N-1}\sin\left(\frac{(l+1)(i+1)\pi}{N+1}\right)\sin\left(\frac{(l-k+1)(j+1)\pi}{N+1}\right) \\ &+\sum_{l=0}^{k-1}\sin\left(\frac{(N-k+l+1)(i+1)\pi}{N+1}\right)\sin\left(\frac{(l+1)(j+1)\pi}{N+1}\right) \\ &+\sum_{l=k}^{N-1}\sin\left(\frac{(l-k+1)(i+1)\pi}{N+1}\right)\sin\left(\frac{(l+1)(j+1)\pi}{N+1}\right)\Big) \end{aligned} \quad (19)$$

which is simplified into

$$a_k(i,j)=A(i,j,k)\cos\left(\frac{(i-j)\pi}{2}\right)+B(i,j,k)\cos\left(\frac{(i+j+2)\pi}{2}\right). \quad (20)$$

Notice that w_{ij} is zero if (i,j) is (odd, even) or (even, odd).

In this manner, the locations of zero elements can be identified and the circular filtering technique can also be established, which are summarized in Table 5. Unfortunately, in the cases of type 2 and type 4 filters, there exist no zero element in the matrix W, implying that the DST domain filtering provides no computational gains. However, for the type 3 filter, the element in the matrix W is zero, if (i,j) is

(even, even) or (odd, odd). Thus, for the type 1 and type 3 filters, half of the elements are zeros. Compared with the DCT domain, the number of zero elements are smaller in the DST domain, indicating that the DST domain filtering is less effective than the DCT domain filtering.

Table 5. Circular filtering in the DST domain. E(even number), D(odd number)

t	zero elments	$Y(m)$
1	$i=E,\ j=D$	$\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j)}X(2j)$, E
	$i=D,\ j=E.$	$\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j+1)}X(2j+1)$, $D.$
3	$i=E,\ j=E$	$\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j)}X(2j)$, D
	$i=D,\ j=D.$	$\sum_{j=0}^{\frac{N}{2}-1} w_{m(2j+1)}X(2j+1)$, $E.$
2 4	no elements	$\sum_{j=0}^{N-1} w_{mj}X(j)$

3. Hartley transform domain

In the Hartley transform domain, the relations for the coefficients $a_k(i,j)$'s are easily obtained in the closed forms. From Table 3, the basis vector t_i for the Hartley transform is given by

$$\begin{aligned} t_i &= \sqrt{\frac{1}{N}}\left(\cos\left(\frac{2\pi il}{N}\right)+\sin\left(\frac{2\pi il}{N}\right)\right) \\ &= \sqrt{\frac{1}{N}}\, cas\left(\frac{2\pi il}{N}\right). \end{aligned} \tag{21}$$

So, it is easy to show that the circular shift is represented by

$$\tilde{t}_{il^*} = t_{i(l-k)}. \tag{22}$$

Also, the inner product between t_i and $\tilde{t}_{j^*}$ is simplified as

$$\begin{aligned} t_i \cdot \tilde{t}_{j^*} &= t_i \cdot t_{j(l-k)} \\ &= \begin{cases} \cos\left(\frac{2ik\pi}{N}\right) & , i=j \\ \sin\left(\frac{2ik\pi}{N}\right) & , i+j=N,\ i\neq j \\ 0 & , otherwise. \end{cases} \end{aligned} \tag{23}$$

From (23), the linear combination coefficients $a_k(i,j)$'s are easily obtained for all filter types. For example, in the type 1 filter, the relation is given by

$$\begin{aligned} a_k(i,j) &= t_i \cdot \tilde{t}_{j^*} + \tilde{t}_{i^*} \cdot t_j \\ &= \begin{cases} 2\cos\left(\frac{2\pi ik}{N}\right) & , i=j \\ 0 & , otherwise. \end{cases} \end{aligned} \tag{24}$$

Note that the relation (24) indicates that the matrix W is diagonal. Thus, only N multiplications are required to carry out the circular filtering.

In the similar manner, the linear combination coefficients for other types are easily derived. The results are shown in Table 6. For the type 3 filter, only N multiplications are required to obtain the filtered output vector, while for the type 2 and type 4 filters, about $2N$ multiplications are required.

Table 6. Circular filtering in the Hartley transform domain.

type	$a_k(i,j)$
1	$2\cos\left(\frac{2\pi ik}{N}\right)$, $i=j$
	0 , $otherwise.$
2	$2\cos\left(\frac{(2k+1)i\pi}{N}\right)\cos\left(\frac{\pi i}{N}\right)$, $i=j\neq\frac{N}{2}$
	$-2\cos\left(\frac{(2k+1)i\pi}{N}\right)\sin\left(\frac{\pi i}{N}\right)$, $i+j=N$
	0 , $otherwise.$
3	$2\sin\left(\frac{2\pi ik}{N}\right)$, $i+j=N$
	0 , $otherwise.$
4	$2\sin\left(\frac{(2k+1)i\pi}{N}\right)\sin\left(\frac{\pi i}{N}\right)$, $i=j\neq 0$
	$2\sin\left(\frac{(2k+1)i\pi}{N}\right)\cos\left(\frac{\pi i}{N}\right)$, $i+j=N$
	0 , $otherwise.$

4. Comparison

In the cases of Hadamard, Haar, and Slant Transforms, the closed forms for the basis vectors are not easy to obtain, since their transformation matrices are determined recursively. So, to inspect the structure of their filtering matrices, first we should calculate the transformation matrix T with the specified block size N. Then, from the calculated basis vectors, all the linear combination coefficients $a_k(i,j)$'s are obtained, according to the relations given in Table 2, and the zero elements are then easily identified.

In Figure 2, for N=8, the matrix W's for six orthogonal transforms are shown as an example. In the Figure, 'o' represents the position of zero elements. It is noted that the positions of the zero elements are fixed, if the transformation and the filter type are given. As discussed previously, if the matrix W is sparse, then the generalized linear filtering saves the computational complexity, compared to the time domain

filtering. In this context, the Hartley transform provides the best computational gain, followd by the DCT and the Hadamard transform. In the case of the DST, the generalized linear filtering is not so efficient as the time domain filtering.

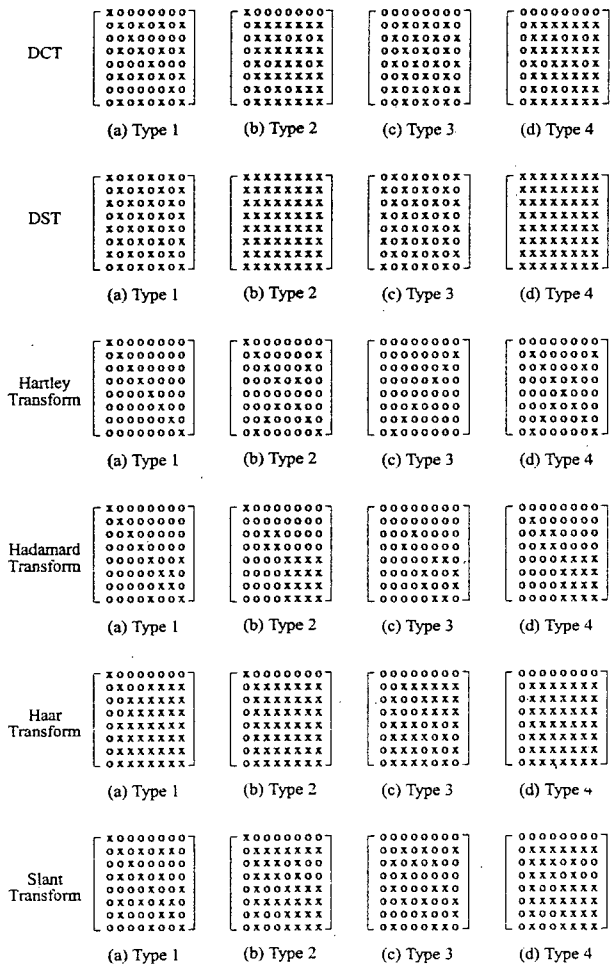

Fig. 3. The from of transform domain filtering matrices.

The comparison of computational complexity among each transform domain filtering is presented in Table 7, where α represents the number of required additions per block, and μ represents the number of required multiplications per block, respectively. Note that in case of the Hartley transform, at most $2N$ multiplications are required for circular filtering, making the generalized linear filtering very effective. For the DCT, the required multiplications with the type 1 filter are about $\frac{N^2}{4}$, which is proportional to the block size N. But, the computational efficiency improves as the block size decreases. For the DST, since the computational loads are much higher than the DCT, the computational savings are not expected.

The sparseness of the matrix W in the proposed filtering technique can be explained as follows. As is discussed previously, the linear combination coefficients $a_k(i,j)$'s, which are obtained by the basis vectors, characterize the elements in the matrix W. The orthogonality of the basis vectors and the symmetry of the filters make many combination coefficients zero, yielding the sparse W matrix. In the generalized linear filtering, it also should be noted that the computational complexity is not dependent on the length of filters, while the computational complexity of the direct time domain processing increase as the length of the filters increase. Thus, the generalized linear filtering is advantageous for the filter of large length.

Table 7. Comparison of required computations.

type		DFT domain	DCT domain	DST domain	Hartley domain
1	α	0	$\frac{N+2}{4}N$	$(\frac{N}{2}-1)N$	0
	μ	N	$\frac{N+2}{4}N$	$\frac{N}{2}N$	N
2	α	0	$\frac{3(N-2)}{4}N$	$(N-1)N$	$N-2$
	μ	N	$\frac{3N-2}{4}N$	N^2	$2N-3$
3	α	0	$(\frac{N}{2}-2+\frac{1}{N})N$	$(\frac{N}{2}-1)N$	0
	μ	N	$(\frac{N}{2}-1)N$	$\frac{N}{2}N$	$N-2$
4	α	0	$\frac{3(N-2)}{4}N$	$(N-1)N$	$N-2$
	μ	N	$\frac{3N-2}{4}N$	N^2	$2N-3$

IV. Decimation and Interpolation in orthogonal transform domain

In this section, we shall discuss about one specific application of the circular filtering technique in orthogonal transform domain : the decimation(subsampling) and interpolation(upsampling). For the sake of simplicity, we shall consider only 2:1 decimation and 1:2 interpolation in this paper. However, the technique can be easily extended to M:1 decimation and 1:M interpolation, respectively.

Figure 4 shows the block diagram for the transform domain decimation.

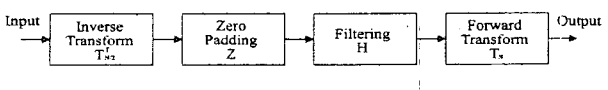

Fig. 4. The procedure of transform domain decimation scheme.

The input and output are N×1 and $\frac{N}{2}\times 1$ vectors of transformed coefficients, respectively. The relation between the input and the output can be expressed as

$$\underline{Y}= T_{\frac{N}{2}} S H T_N^T \underline{X}, \tag{25}$$

where S is the $\frac{N}{2}\times N$ selection matrix, given by

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ \cdots & & & & & \\ & & \cdots & & & \\ 0 & 0 & \cdots & 0 & 1 & 0 \end{bmatrix},$$

$T_{\frac{N}{2}}$ is the $\frac{N}{2}\times\frac{N}{2}$ transform matrix, and T_N is the $N\times N$ transform matrix, respectively.

Now, we shall define an $\frac{N}{2}\times N$ transform domain decimation matrix D as

$$D= T_{\frac{N}{2}} S H T_N^T. \tag{26}$$

To analyze the decimation, let us pay our attention to the elements of the matrix D. As in the generalized linear filtering, the sparseness of matrix D determines the efficiency of the transform domain decimation.

For convenience, let us define even selection vector operator ${}_e\underline{f}$ and odd selection vector operator ${}_o\underline{f}$, respectively, as

$$\begin{cases} {}_e f_l = f_{(2l)}, \\ {}_o f_l = f_{(2l+1)}. \end{cases} \tag{27}$$

Then, the element of matrix D, d_{ij} can be derived in the same manner as w_{ij}, which are shown in Table 8. Notice that the filters used for decimation is a lowpass filter. So, among linear phase filters, type 1 and type 2 are possible choices.

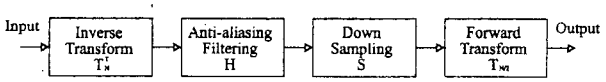

Fig. 5. The procedure of transform domain interpolation scheme.

In the same manner, we can derive the interpolation schemes in orthogonal transform domain. In Figure 5, the $N\times\frac{N}{2}$ transform domain interpolation matrix U is expressed as

$$U= T_N H Z T_{\frac{N}{2}}^T, \tag{28}$$

where Z is the $N\times\frac{N}{2}$ zero padding matrix, which is the transpose of the matrix S.

Table 8 lists the properties of the elements u_{ij}'s of the matrix U with the linear phase filters.

Table 8. The elements of matrix D. E(even), D(odd).

t	d_{ij}	$a_k(i,j)$	
1	$\sum_{k=0}^{K} a_k(i,j)h(k)$	$\underline{g}_i \cdot {}_e\underline{t}_j$	$,k=0$
		$\underline{g}_i \cdot {}_e\tilde{\underline{t}}_j^{\frac{k}{2}} + \tilde{\underline{g}}_i^{\frac{k}{2}} \cdot {}_e\underline{t}_j$	$,k=E$
		$\underline{g}_i \cdot {}_o\tilde{\underline{t}}_j^{\frac{k+1}{2}} + \tilde{\underline{g}}_i^{\frac{k-1}{2}} \cdot {}_o\underline{t}_j$	$,k=D$
2	$\sum_{k=0}^{K-1} a_k(i,j)h(k)$	$\underline{g}_i \cdot {}_e\tilde{\underline{t}}_j^{\frac{k}{2}} + \tilde{\underline{g}}_i^{\frac{k}{2}} \cdot {}_o\underline{t}_j$	$,k=E$
		$\underline{g}_i \cdot {}_o\tilde{\underline{t}}_j^{\frac{k+1}{2}} + \tilde{\underline{g}}_i^{\frac{k+1}{2}} \cdot {}_e\underline{t}_j$	$,k=D$

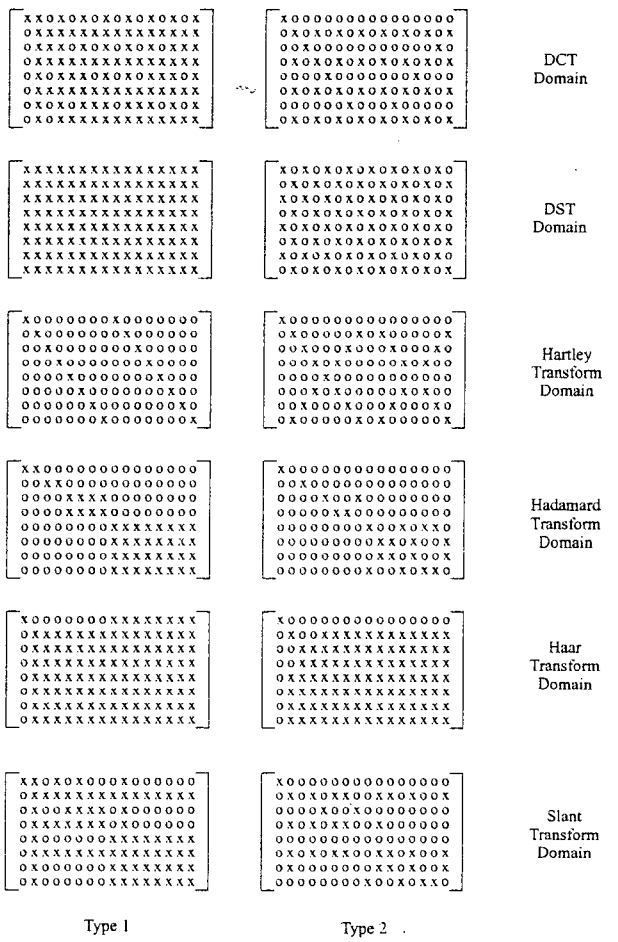

Fig. 6. The structure of matrix D.

In Figure 6, for N=16, the structure of the matrix D for six orthogonal transforms are shown as an example. Note that there exist many zero elements in the matrices, which

determine the efficiency of the transform domain decimation. In the DCT domain, with the type 1 filter, there are small numbers of zero elements in the matrix D. However, the matrix D is very sparse for the type 2 filter, yielding the computational savings. More specifically, about $\frac{3}{4}$ of the elements in the matrix D are zeros, indicating that about $\frac{N^2}{8}$ multiplications are required for the decimation. In the case of the Hartley transform, the computational efficiency is distinct. It is seen that only N multiplications are required for the type 1 filter and less than $2N$ multiplications for the type 2 filter. The sparseness of the matrix D for the Hadamard transform is also noticeable. With the type 2 filter, more computational gain is achieved than with the type 1 filter.

The matrix U has the same number of zero elements as the matrix D, which can be easily verified. So, the computational complexity for the transform domain interpolation are same as that of the decimation.

The transform domain decimation/interpolation technique can be easily incorporated into the SBC-DCT coding techniques[8, 12]. The SBC-DCT coding technique is one of the popular image compression techniques, since it alleviates the visually annoying blocking effect and provides high energy packing efficiency. In general, the SBC-DCT coding technique is composed of five steps: analysis filtering, the 8 $\times$8 DCT, quantization, 8$\times$8 IDCT, and synthesis filtering. Note that in the first and final steps, the decimation and interpolation are carried out, respectively. However, for the purpose of layered coding, 16$\times$16 DCT together with the blockwise subband decomposition has been also proposed[10], which is known to be very effective in the prioritized ATM networks. We are concerned with the blockwise subband decomposition technique in this paper. Instead of the time domain processing, the DCT domain decimation and interpolation can be utilized, combined with the 16$\times$16 DCT as shown in Figure 7. First, the input image is transformed by the 16$\times$16 DCT, followed by the DCT domain decimation as an analysis step, which yields 2$\times$2 subband coefficients. Then each band is quantized, respectively. In the decoding, the DCT domain interpolations are carried out as synthesis filtering, followed by the 16$\times$16 IDCT.

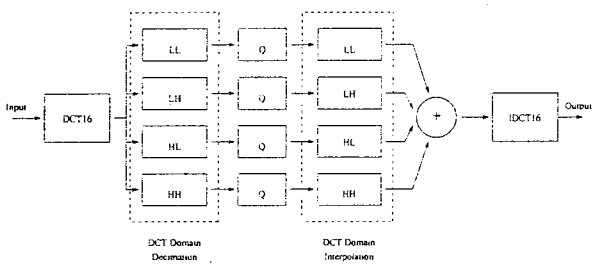

Fig. 7. Subband coding using DCT domain decimation/ interpolation.

Table 10 shows the numbers of required multiplications and additions per pixel in each step. In Table 10, the DCT's are assumed to be implemented by a fast algorithm[11]. The Johnston's 12-tab filters[9] are chosen in the analysis and synthesis step. Compared with the conventional implementation, the computational complexity is slightly increased in the DCT and the IDCT, but considerable computational savings can be achieved in the analysis/ synthesis filtering step by utilizing the DCT domain decimation/interpolation technique. Conclusively, about 30 percents of the computational complexity can be saved.

Table 9. Elements of the matrix U. E(even), D(odd).

t	u_{ij}	$a_k(i,j)$
1	$\sum_{k=0}^{K} a_k(i,j)h(k)$	${}_e\underline{t}_i \cdot \underline{g}_j$, $k=0$ ${}_e\underline{t}_i \cdot \check{\underline{g}}_j^{\frac{k}{2}} + {}_e\check{\underline{t}}_i^{\frac{k}{2}} \cdot \underline{g}_j$, $k=E$ ${}_o\underline{t}_i \cdot \check{\underline{g}}_j^{\frac{k-1}{2}} + {}_o\check{\underline{t}}_i^{\frac{k+1}{2}} \cdot \underline{g}_j$, $k=D$
2	$\sum_{k=1}^{K+1} a_k(i,j)h(k)$	${}_e\underline{t}_i \cdot \check{\underline{g}}_j^{\frac{k}{2}} + {}_o\check{\underline{t}}_i^{\frac{k}{2}} \cdot \underline{g}_j$, $k=E$ ${}_o\underline{t}_i \cdot \check{\underline{g}}_j^{\frac{k-1}{2}} + {}_e\check{\underline{t}}_i^{\frac{k-1}{2}} \cdot \underline{g}_j$, $k=D$

Table 10. Required numbers of multiplications and additions in subband coding schemes.

		DCT	Analysis	Synthesis	IDCT	Sum	Reduction
conventional	μ	1.5	12	24	1.5	39	-
	α	7.3	22	22	7.3	58.6	-
proposed	μ	2	12.3	12.3	2	28.6	27.7 %
	α	9.8	10.3	10.3	9.8	40.2	31.9 %

V. Conclusion

In this paper, based on the generalized linear filtering, we have provided an exact circular filtering technique in the orthogonal transform domains. The computational efficiency of the proposed filtering technique has also been intensively investigated. The results show that the circular filtering can be efficiently implemented in the Hartley transform, the DCT and Hadamard transform domains. As an application of the filtering, decimation and interpolation techniques in orthogonal transform domains have been presented, which is suitable to the SBC-DCT coding techniques.

The conventional DCT filtering techniques[1,2,3] have not implemented the exact circular filtering, since they are based on the approximation. However, the proposed technique

implements the circular filtering in orthogonal transform domains exactly. Moreover, compared to the time domain processing, this technique achieves much computational savings. It will furnish new applications for the orthogonal transforms, such as applications to the perfect filter banks and image compression techniques. However, further studies are necessary to make use of the circular filtering technique.

References

[1] W. H. Chen and S. C. Fralic, "Image enhancement using cosine transform filtering," *in Proc. Symp. on Current Math. Problems in Image Science*, Montery, CA, Nov. 1976, pp. 186-192.

[2] K. N. Ngan and R. J. Clarke, "Lowpass filtering in the cosine transform domain," *in Proc. Internat. Conf. on Communi*, Seattle, WA, June 1980, pp. 31.7.1-31.7.5

[3] B. Chitprasert and K. R. Rao, "Discrete cosine transform filtering," *Signal Processing*, vol. 19, no. 3, pp. 233-245, Mar. 1990.

[4] J. B. Lee and B. G. Lee, "Transform domain filtering based on pipelining structure," *IEEE Trans. Signal Processing*, vol. 40, no. 8, pp. 2061-2064, Aug. 1992.

[5] S. A. Martucci and R. M. Mersereau, "The symmetric convolution approach to the nonexpensive implementation of FIR filter backs for images," *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* 1993. vol 5. pp 65-68.

[6] W. K. Pratt, *Digital Image Processing*, New York : John Wiley & Sons, 1991.

[7] A. V. Oppenheim and R. W. Schafer, *Discrete-time Signal Processing*, Englewood Cliffs, NJ. : Prentice Hall, 1992.

[8] H.Gharavi, "Subband coding algorithms for video applications: Videophone to HDTV-conferencing," *IEEE Trans. Cir. and Syst. for Video Technology*, vol. 1, no. 2, pp. 174-183, June 1991.

[9] J. D. Johnston, "A filter family design for use in quadrature mirror filter banks," *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Denver, 1980, pp. 281-291.

[10] B. DeCleene and H. Sorensen, "The application of subband coding in MPEG for prioritized ATM networks," *in Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing* 1994. vol. 5, pp. 433-436.

[11] N. I. Cho and S. U. Lee, "Fast algorithm and implementation of 2-D discrete cosine transform," *IEEE Trans. on Circuit and Systems*, vol. 38, no. 3, pp. 297-305, Mar. 1991.

[12] H. Paek, R. C. Kim, and S. U. Lee, "On the motion compensated transform coding technique employing sub-band decomposition," *in proc. SPIE Vol. 1818 Visual Communications and Image Processing*, 1992, pp. 253-264.

Bong Seop Song received the B.S. and M.S. degrees form Seoul National University, Seoul, Korea, in 1992 and 1994, respectively in electrical engineering. He is currently working toward the Ph.D. degree at Seoul National University. His research interests is in digital signal processing, including image processing and computer vision.

Sang Uk Lee received the B.S. degree from Seoul National University in 1973, the M.S. degree from Iowa State University in 1976, and the Ph.D. degree from the University of Southern California, Los Angeles, in 1980, all in electrical engineering. In 1980, he was with General Electric Company, Lynchburg, VA, and in 1981-1983, he was a member of Technical Staff, M/A-COM Research Center, Rockville, MD. In 1983, he joined the school of Electrical Engineering at Seoul National University as an Assistant Porfessor, where he is now a Professor. His current research interests are in the areas of image processing, digital communication, and computer vision. Dr. Lee is a member of Phi Kappa Phi.