FACTORIZATION OF NONUNIFORM BLOCK ORTHOGONAL TRANSFORMS

Takayuki Nagai　　*Shigeki Obata*　　*Masaaki Ikehara*

Department of Electrical Engineering, Faculty of Science and Technology, Keio University
3-14-1, Hiyoshi, Kohoku-ku, Yokohama, 223 JAPAN

ABSTRACT

Block orthogonal transforms (BOT's) are commonly used for lots of applications. Conventional BOT's are based on uniform filter banks, however, nonuniform BOT's are often superior to uniform ones. In this research, we investigate the factorization of nonuniform BOT's which does not involve the tree structure. Therefore, optimal nonuniform BOT's are available in the sense of transform coding gain. Some design examples are included to confirm our theory. We also apply the nonuniform BOT to the transform image coding.

1. INTRODUCTION

Block orthogonal transforms have a lot of applications such as image compression, pattern recognition and so on, and some BOT's have been reported. Almost all of these transforms divide the signal uniformly in the frequency-domain, and KLT is optimal in this case. In other hands, BOT's based on nonuniform filter banks have been proposed, and it is shown that nonuiform BOT's have a possibility to outperform the conventional uniform BOT's [2]. However, the proposed nonuniform BOT's in Ref.[2] are not optimal since the tree structure is involved in constructing them.

In this paper, we are interested in nonuniform BOT's and investigate how to construct them directly. This direct factorization has following three advantages over the tree structure:

1. One can control the characteristic of each filter directly. This means that an optimal nonuniform BOT will be obtained.
2. The number of free parameters increases for some cases. Actually, it depends on the sampling factor.
3. There are some frequency divisions which are impossible to be implemented by any tree structures, while they can be constructed by the direct approach [3].

In order to factorize nonuniform BOT's, the transformation, which reduces a nonuniform filter bank to an equivalent uniform one, is used. Then we consider how to factorize this transformed orthogonal matrix. Some design examples are included to show the validity of our proposed method. Further, our proposed nonuniform BOT's are tested on the transform image coding.

2. NONUNIFORM BOT

2.1. Nonuniform BOT and equivalent uniform BOT

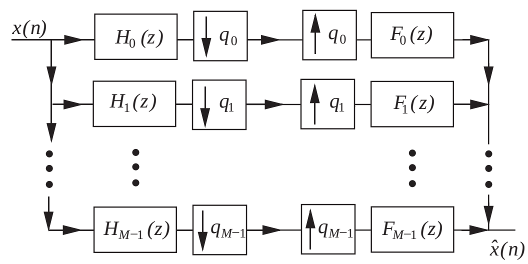

Figure 1. Nonuniform BOT's implementation using NUFB.

Fig.1 shows the nonuniform BOT implemented by the nonuniform filter bank (NUFB), where q_i is a nonnegative integer. The sampling factor is expressed as $[q_0 \ \cdots \ q_{M-1}]$ in this paper. The NUFB is assumed to be critically sampled, that is, $\sum_{i=0}^{M-1} 1/q_i = 1$ and all filters have linear phase. The lengths of i-th analysis and synthesis filters are both q_i. Moreover, each synthesis filter can be written

$$F_k(z) = z^{-q_k+1} H_k(z^{-1}), \quad k = 0, \cdots, M-1 \qquad (1)$$

since the filter bank is orthogonal.

Now let Q be a least common multiple of q_i ($i = 0, \cdots, M-1$). Fig.1 can be transformed to a Q-channel uniform filter bank shown in Fig.2 [2][3]. The analysis filter $G_k(z)$ can be expressed as follows:

$$G_{(\sum_{i=0}^{k} r_i)-r_k+u_k}(z) = z^{-u_k q_k} H_k(z), \qquad k = 0, \cdots, M-1 \qquad (2)$$

where $r_k = Q/q_k$ and $u_k = 0, 1, \cdots, r_k - 1$. Thus one can define a transform matrix of the nonuniform BOT. Naturally, it corresponds to the poly phase component matrix of Fig.2 and is orthogonal. Unlike the uniform case, it is not a trivial task to factorize nonuniform BOT's since they have a special form (the transform matrix contains shifted version of the same filter). The transform matrix of [8 8 8 8 4 4] is

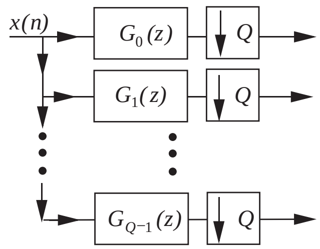

Figure 2. Equivalent uniform filter bank of Fig.1.

shown below for example:

$$\boldsymbol{T}_{8,6}=\begin{bmatrix} a & b & c & d & d & c & b & a \\ e & f & g & h & -h & -g & -f & -e \\ i & j & k & l & l & k & j & i \\ m & n & o & p & -p & -o & -n & -m \\ q & r & r & q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q & r & r & q \\ s & t & -t & -s & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s & t & -t & -s \end{bmatrix}. \quad (3)$$

The symmetry polarity of each filter is chosen according to the corresponding tree structure. The objective of this research is to factorize the orthogonal matrix such as Eq.(3).

2.2. Coding gain of nonuniform BOT

Transform coding gain is a good criterion to measure the efficiency of a orthogonal transform in the context of signal compression. It's the ratio of the AM to GM of the variances of the subband signals[1]

$$G_{TC}=\frac{\sum_{K=0}^{M-1}\sigma_{y_k}^2}{M\left(\prod_{K=0}^{M-1}\sigma_{y_k}^2\right)^{1/M}} \quad (4)$$

where $\sigma_{y_k}^2$ denotes the variance of the k-th subband. G_{TC} in Eq.(4) can be also applicable to any nonuniform BOT's since they can be transformed to the equivalent Q-channel uniform BOT's. In this case M in Eq.(4) has to be changed to Q.

3. FACTORIZATION OF NONUNIFORM BOT

3.1. Factorization

In this section we describe how to factorize nonuniform BOT's. Only the case where Q is even is considered here since the size of the uniform BOT is generally even. For a convenience of the explanation, we concentrate on the sampling factor [8 8 8 8 4 4] whose transform matrix is given in Eq.(3). We first pay attention to the fact that the sum of 5-th and 6-th rows yields a symmetric row and their difference produces an anti-symmetric one. Further, the sum of 7-th and 8-th rows yields an anti-symmetric row and their difference produces a symmetric one. Therefore the following matrix $\boldsymbol{S}_0$ is multiplied and then, a permutation matrix $\boldsymbol{P}$ which changes the order of rows is multiplied to obtain the following transform matrix:

$$\boldsymbol{P}\boldsymbol{S}_0\boldsymbol{T}_{8,6}=\begin{bmatrix} \boldsymbol{A} & \boldsymbol{A}\boldsymbol{J}_4 \\ \boldsymbol{B} & \boldsymbol{B}\boldsymbol{J}_4 \\ \boldsymbol{C} & -\boldsymbol{C}\boldsymbol{J}_4 \\ \boldsymbol{B} & -\boldsymbol{B}\boldsymbol{J}_4 \end{bmatrix} \quad (5)$$

where $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ denote 2×4 matrices and

$$\boldsymbol{S}_0=\begin{bmatrix} \boldsymbol{I}_4 & \boldsymbol{0}_4 \\ \boldsymbol{0}_4 & \begin{matrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & 0 \\ 0 & 0 & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{matrix} \end{bmatrix}. \quad (6)$$

Moreover, $\boldsymbol{B}$ can be written as

$$\sqrt{2}\boldsymbol{B}=\begin{bmatrix} q & r & r & q \\ s & t & -t & -s \end{bmatrix}=\begin{bmatrix} \boldsymbol{a} & \boldsymbol{a}\boldsymbol{J}_2 \\ \boldsymbol{b} & -\boldsymbol{b}\boldsymbol{J}_2 \end{bmatrix}. \quad (7)$$

$\boldsymbol{I}_k$ and $\boldsymbol{J}_k$ represent identity and counter-identity matrices of size k, respectively. Since $\boldsymbol{P}\boldsymbol{S}_0\boldsymbol{T}_{8,6}$ in Eq.(5) has a completely same characteristic as uniform BOT, $\boldsymbol{R}_8$ is multiplied:

$$\boldsymbol{P}\boldsymbol{S}_0\boldsymbol{T}_{8,6}\boldsymbol{R}_8=\begin{bmatrix} \sqrt{2}\boldsymbol{A} & \boldsymbol{0}_4 \\ \sqrt{2}\boldsymbol{B} & \\ \boldsymbol{0}_4 & \sqrt{2}\boldsymbol{C} \\ & \sqrt{2}\boldsymbol{B} \end{bmatrix}=\begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{0}_4 \\ \boldsymbol{0}_4 & \boldsymbol{D}_1 \end{bmatrix} \quad (8)$$

where $\boldsymbol{R}_i$ denotes $i\times i$ orthogonal matrix such as

$$\boldsymbol{R}_i=\frac{1}{\sqrt{2}}\begin{bmatrix} \boldsymbol{I}_{i/2} & \boldsymbol{I}_{1/2} \\ \boldsymbol{J}_{i/2} & -\boldsymbol{J}_{i/2} \end{bmatrix}. \quad (9)$$

Thus, the factorization problem becomes finding two 4×4 orthogonal matrices $\boldsymbol{D}_0$ and $\boldsymbol{D}_1$. It is well known that 4×4 orthogonal matrix can be represented as a product of six plane rotation matrices. However, the lower half of $\boldsymbol{D}_1$ coincides with that of $\boldsymbol{D}_1$, namely $\sqrt{2}\boldsymbol{B}$, the rotation angles have to be the same except for one. Moreover, $\boldsymbol{D}_0$ can be written as:

$$\boldsymbol{D}_0=\begin{bmatrix} \sqrt{2}\boldsymbol{A} & \\ \boldsymbol{a} & \boldsymbol{a}\boldsymbol{J}_2 \\ \boldsymbol{b} & -\boldsymbol{b}\boldsymbol{J}_2 \end{bmatrix} \quad (10)$$

where $\boldsymbol{a}$ and $\boldsymbol{b}$ are 1×2 vectors. $\boldsymbol{D}_0$ can be further decomposed as follows:

$$\boldsymbol{D}_0=\begin{bmatrix} \hat{\boldsymbol{A}} & \\ \sqrt{2}\boldsymbol{a} & \boldsymbol{0}_{1,2} \\ \boldsymbol{0}_{1,2} & \sqrt{2}\boldsymbol{b} \end{bmatrix}\boldsymbol{R}_4^T \quad (11)$$

where $\hat{\boldsymbol{A}}$ is a 2×4 matrix and $\boldsymbol{0}_{1,2}$ represents a 1×2 null vector. Therefore $\boldsymbol{D}_0$ can be factorized as

$$\begin{aligned}\boldsymbol{D}_0 &= \begin{bmatrix} c_0 & s_0 & 0 & 0 \\ -s_0 & c_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} c_1 & 0 & 0 & s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s_1 & 0 & 0 & c_1 \end{bmatrix} \\ &\times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & s_2 & 0 \\ 0 & -s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\boldsymbol{S}_1\boldsymbol{R}_4^T \end{aligned} \quad (12)$$

where c_i, s_i denote $\cos\theta_i$ and $\sin\theta_i$, respectively, and

$$\boldsymbol{S}_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Although $\boldsymbol{D}_1$ can be parameterized as above, two parameters θ_1 and θ_2 should be identical with those of $\boldsymbol{D}_0$. Accordingly the number of parameters for this nonuniform BOT is 4. From the above discussion, we can summarize that the transform matrix can be factorized as follows (Fig.3):

$$\boldsymbol{T}_{8,6} = \boldsymbol{S}_0 \boldsymbol{P}^T \begin{bmatrix} \boldsymbol{D}_0 & \boldsymbol{0}_4 \\ \boldsymbol{0}_4 & \boldsymbol{D}_1 \end{bmatrix} \boldsymbol{R}_8^T. \quad (13)$$

The similar factorization is applicable for other sampling factor cases.

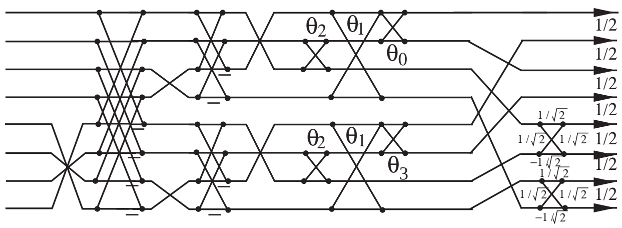

Figure 3. Implementation of the nonuniform BOT.

3.2. Number of free parameters

As we described, the number of parameters (rotation angles) of our proposed BOT's is more than that of BOT's based on tree structure in some cases. We made a comparison between these for $Q = 8$ and $Q = 10$. Tab.1 shows the results. For the tree structure, some different conbinations of uniform BOT's are possible to construct a nonuniform BOT. For example, there are two ways to construct [8 8 8 8 2], namely harr+4channel uniform BOT and the combination of harrs. The former has 2 free parameters while the latter has no free parameters. In the comparison, we consider the structure which yeilds the maximum number of parameters.

From the results, we conjectured that the number of parameters will increase in the case where $M \geq Q/2 + 2$. The direct approach yields a BOT with better characteristic even though they have equal number of parameters.

4. SIMULATION RESULTS

4.1. Design examples

We designed the nonuniform BOT whose transform matrix is represented in Eq.(3). Fig.4(a) shows the frequency response of the BOT for AR(1) (Transform coding gain was maximized ($\rho_1 = 0.95$)). The BOT shown in Fig.4(b) was optimized for AR(2) process with $\rho_1 = 0.0$, $\rho_2 = 0.56$ (low+high frequency signal). The nonuniform BOT's based on tree structure (DCT4+harr) are given in Fig.4(c) and (d) for the comparison purpose. One can confirm that the direct factorization provides better results. Tab.2 shows the comparison of transform coding gain for AR(1) process ($\rho_1 = 0.95$).

Table 1. Comparison of the number of parameters for $Q = 8$ and $Q = 10$.

Q	M	Sampling factor	Direct	Tree
8	8	Uniform (KLT)	12	–
8	7	[8 8 8 8 8 8 4]	7	2
8	6	[8 8 8 8 4 4]	4	2
8	5	[8 8 8 8 2],[8 8 4 4 4]	2	2
8	4	[8 8 4 2]	0	0
10	10	Uniform (KLT)	20	–
10	9	[10 10 10 10 10 10 10 10 5]	13	4
10	8	[10 10 10 10 10 10 5 5]	9	4
10	7	[10 10 10 10 5 5 5]	5	4
10	6	[10 10 5 5 5 5] [10 10 10 10 10 2]	4	4

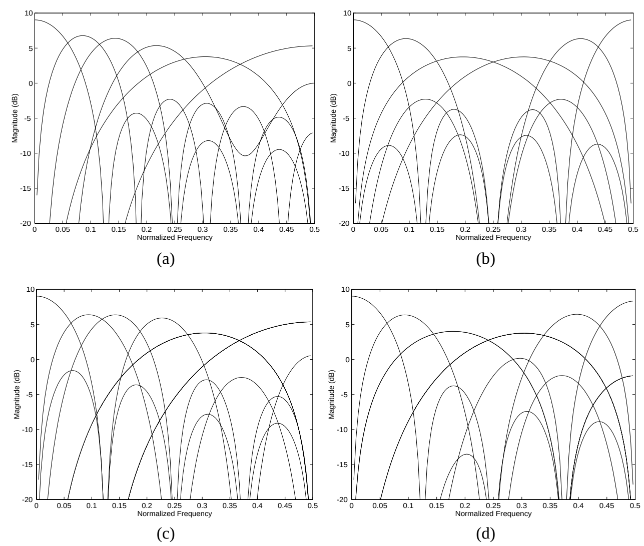

Figure 4. Frequency responses of the design examples.

4.2. Application in image coding

Nonuniform BOT's are tested on the transform image coding. The same transform-based (JPEG-like) coder is used for all cases. 512×512 "Lena" was used as the input image. Especially for low bit rate, proposed BOT improves the coding performance considerably. The results for compressed at 0.1 bpp are listed in Tab.3. Fig.5 shows zoom-in portions of the original and reconstructed images.

5. CONCLUSION

In this research we studied the theory and the factorization of nonuniform BOT's. By using our proposed factorization, optimal nonuniform BOT's can be obtained in the sense of transform coding gain. Some design examples imply this fact. We also apply the nonuniform BOT to the transform image coding. The results show that the nonuniform BOT is efficient in the application. The extension to LOT is left for the future research.

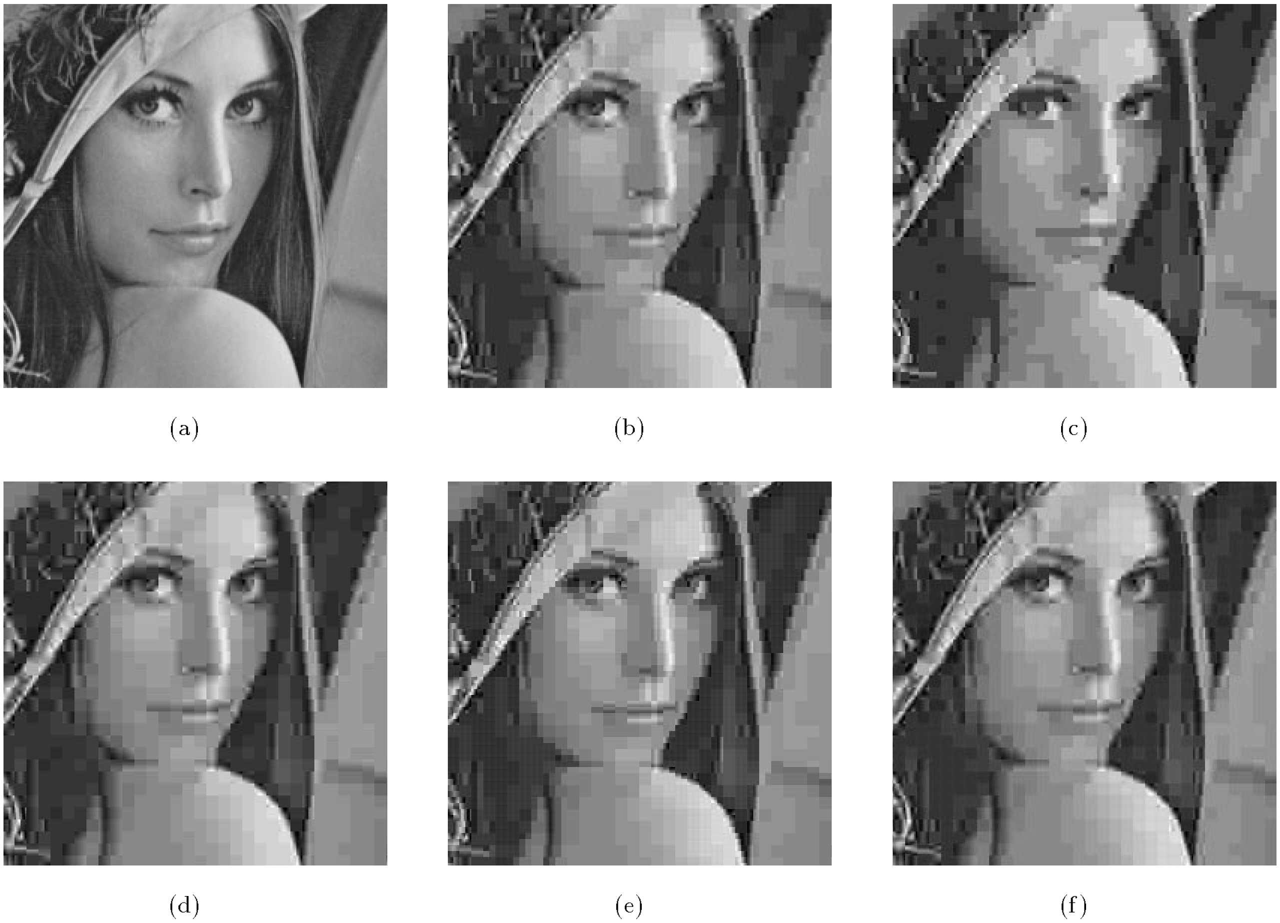

Figure 5. Original and reconstructed images. (a)Original image. (b)8-point DCT. (c)6-point DCT. (d)Nonuniform BOT (Direct $Q = 8$, $M = 6$). (e)Nonuniform BOT (DCT4+harr $Q = 8$, $M = 6$). (f)Nonuniform BOT (Direct $Q = 8$, $M = 7$).

Table 2. Coding gain comparison (dB).

	Direct	Tree
$Q = 8, M = 7$	8.8246	8.3411
$Q = 8, M = 6$	8.7128	8.3411
DCT8	8.8259	–
KLT8	8.8259	–
DCT6	8.4072	–
KLT6	8.425	–

Table 3. Coding performance comparison.

	PSNR	Max Error	Taps per Sample
$Q = 8, M = 7$ (Direct)	28.04	113	7
$Q = 8, M = 7$ (DCT4+harr)	24.00	222	7
$Q = 8, M = 6$ (Direct)	27.89	115	6
$Q = 8, M = 6$ (DCT4+harr)	27.12	132	6
DCT8	28.03	116	8
DCT6	26.52	136	6

REFERENCES

[1] P.P.Vaidyanathan, Multirate Systems and Filter Banks, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[2] A. Makur, "BOT's Based on Nonuniform Filter Banks", IEEE Tras. Signal Processing vol.44, NO.8, pp1971-1981, August.1996.

[3] J.Kovacevic and M.Vetterli, "Perfect Reconstruction Filter Banks with Rational Sampling Facters", IEEE Trans. Signal Processing, vol.41, NO.6, pp2047-2066, June 1993.