PROCESSOR ARCHITECTURE FOR EXTENDED LAPPED TRANSFORM

David Akopian and Jaakko Astola
Signal Processing Laboratory, Tampere University of Technology
P.O.Box 553, FIN-33101, Tampere, Finland
e-mails: prog@cs.tut.fi, jta@cs.tut.fi

ABSTRACT

This paper is devoted to implementation of Extended Lapped Transform (ELT), which is among the most efficient factorization methods for paraunitary filterbanks. First we utilize the special form of the matrices in the part of factorization of ELT to process data at input data rate in a pipelined structure with minimal number of processor elements without inserting additional delays. Next we suggest an algorithm for DCT-IV transform, the other part of ELT factorization, with a constant geometry structure suitable for the use of perfect-shuffle network.

1 INTRODUCTION

Multirate filter banks and lapped transforms are powerful tools in modern digital signal processing. Although studied independently in the past, both represent the same concept [1].

In this work we consider the algorithm that is suitable for modular hardware implementation of extended lapped transform (ELT) [2], which is among the most efficient factorization methods for a paraunitary filter bank. This class of transforms includes modified cosine or modulated lapped transforms (MLT) [1],[3], when the length of filters in the equivalent filter bank is constrained to be equal to twice the number of subbands. For the applications of ELT we refer to the [1], [4], [5].

The fast ELT structure (see Fig.1 and (3),(4)) is based on DCT-IV block transform plus K butterfly stages. For M-band ELT the computational complexity is of order $MK + M\log_2 M$. For data-serial input architecture based on one arithmetic unit will perform $O(K+\log_2 M)$ operations per sample, thus constraining the input data rate. The maximally parallel structure containing $O(MK+M\log_2 M)$ computational elements has the highest data rate for this algorithm independent on M and K with the delay of $K+\log_2 M$ stages.

We propose a parallel structure with $O(K+M)$ computational elements and the same throughput as the maximally parallel structure.

First, we utilize the special form of the matrices in the first part of the factorization of ELT to process data at input data rate in a pipelined structure with minimal number of processor elements without inserting additional delays. Next we realize directly DCT-IV transform, the other part of ELT factorization, on perfect shuffle network and derive a constant geometry fast algorithm.

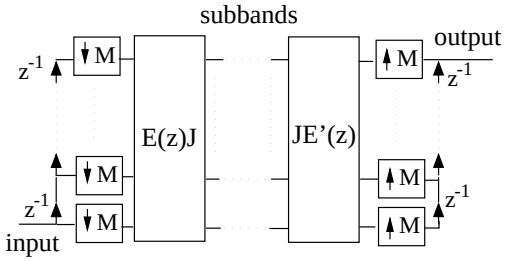

Figure 1: ELT polyphase structure (3),(4)

2 DEFINITIONS

Lapped transforms are defined by the rectangular matrix P with the elements determining the impulse responses of the analysis and synthesis filters, that is,

$$f_k(n) = p_{nk},\ h_k = f_k(NM-1-n) = p_{NM-1-n,k}, \quad (1)$$

where $k = 0, 1, \ldots, M-1,\ n = 0, 1, NM-1$. For ELT the basis functions are defined as

$$p_{nk} = h(n)\sqrt{\frac{2}{M}}\cos\left[\frac{\pi(2k+1)(2n+M+1)}{4N}\right], \quad (2)$$

where $h(n)$ is the low-pass prototype, and the length of the basis functions should be an even multiple of the number of subbands M, i.e., $N = 2K$. K is called overlap factor and for modulated lapped transform we have $K = 1$.

According to fast algorithm for ELT [1] the polyphase component matrix E for the equivalent filter bank can be factorized in the following way

$$EJ = C^{IV} I^* Z_1 D_0 Z_2 D_1 \cdots Z_2 D_{K-1}, \quad (3)$$

$$JE' = D_{K-1} Z'_2 \cdots D_1 Z'_2 D_0 Z'_1 I^* C^{IV} \quad (4)$$

where C^{IV} is the type IV cosine transform [6],

$$Z_i = \begin{bmatrix} z^{-i}I & 0 \\ 0 & I \end{bmatrix}, \quad D_k = \begin{bmatrix} -C_k & S_k J \\ JS_k & JC_k J \end{bmatrix},$$

$$Z'_i = \begin{bmatrix} I & 0 \\ 0 & z^{-i}I \end{bmatrix}, I^* = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix},$$

and $C_k = diag\{\cos(\theta_{0,k})\cos(\theta_{1,k})\cdots\cos(\theta_{M/2-1,k}),\ S_k = diag\{\sin(\theta_{0,k})\sin(\theta_{1,k})\cdots\sin(\theta_{M/2-1,k}),\ \theta$ are the rotation angles and free parameters in the design of an ELT [1], J is the counterdiagonal matrix (ones only in the counter diagonal). See Fig.1 and (3),(4) for analysis and synthesis sections of the ELT.

3 CONSTANT GEOMETRY FAST DCT-IV ALGORITHMS

In [7] we have presented a general method to transfer certain kind of fast algorithms to algorithms with constant geometry (CG) structure. It was indicated that two CG algorithms for Discrete sine transform (DST-IV) can be derived from Wang's DST-IV algorithm [6] and one of them was presented. These algorithms can be applied also to DCT-IV transform because of the relation:

$$C^{IV} = JS^{IV}\tilde{I} \tag{5}$$

where C^{IV}, S^{IV} are the DCT-IV and DST-IV transform matrices accordingly, $\tilde{I} = diag\left((-1)^i\right),\ i = 0,\ldots,M-1$.

The CG DCT-IV algorithm is derived from the one presented in [7] for DST-IV. Denote

$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad R_M = \bigoplus_{0}^{M/4} R, \tag{6}$$

$$F_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad M_1 = I_2 \otimes F_2, \quad M_2 = F_2 \otimes I_2 \tag{7}$$

$$B_{n-i} = \bigoplus_{\tau=1}^{2^{n-2-i}} \bigoplus_{k=0}^{2^i-1} M_2 \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 2d_i^k & 0 \\ 0 & 0 & 0 & 2d_i^k \end{pmatrix} R$$

$$B_1 = \bigoplus_{k=0}^{2^{n-2}-1} M_1 \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 2d_{n-1}^{2k} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2d_{n-1}^{2k+1} \end{pmatrix}$$

$$B_n = \bigoplus_{\tau=1}^{2^{n-1}} \bigoplus_{k=0}^{2^i-1} M_2 \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 2d_0^0 & 0 \\ 0 & 0 & 0 & 2d_0^0 \end{pmatrix}$$

where $n = \log M,\ i = 1,\ldots,n-2,\ d_i^t = d(2^i + t), d(k+t) = \cos[(h_k(t) + 1/2)\frac{\pi}{2k}]$, $h_k(t)$ corresponds to Hadamard reordering [6][7]. Then DCT-IV transform matrix $C_{2^n}^{IV}$ is decomposed as (without normalization factors)

$$C_{2^n}^{IV} = JH^T D^* B_1 \left(\prod_{i=2}^{n-1}(B_i P_{M,2}^T)\right) B_n P_1 \tilde{I}_M \tag{8}$$

where H is the Hadamard reordering matrix, $P_{M,2}(x_0, x_1, \ldots, x_{M-1})^T = (x_0, x_2, \ldots, x_{M-2}, x_1, x_3, \ldots, x_{M-1})^T$ is the unshuffle reordering, $D_S^* = diag\left(\sin[\frac{\pi}{2M}(h_M(i) + \frac{1}{2})]\right), i = 0,\ldots,M-1$.

$$P_1 = \overbrace{P_{M,2}R_M \ldots P_{M,2}R_M}^{n-2 \text{ times}} \tag{9}$$

The efficient bit level generation of indeces according to P_1 with $O(\log M)$ area and delay complexity was suggested in [7].

The other algorithm can be derived in analogous to the previous way. Denote

$$B_{n-i} = \bigoplus_{\tau=1}^{2^{n-1-i}} \bigoplus_{k=0}^{2^{i-1}-1} \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 2d_i^{2k} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2d_i^{2k+1} \end{pmatrix} M_2 R$$

$$B_1 = \bigoplus_{k=0}^{2^{n-2}-1} M_1 \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 2d_{n-1}^{2k} & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 2d_{n-1}^{2k+1} \end{pmatrix} M_2 R$$

$$B_n = \bigoplus_{\tau=1}^{2^{n-1}} \begin{pmatrix} 1 & 1 \\ 0 & 2d_0^0 \end{pmatrix}$$

Now DCT-IV transform matrix $C_{2^n}^{IV}$ (without normalization factors) is decomposed as

$$C_{2^n}^{IV} = JP_H^T D^* \left(\prod_{i=1}^{n-1}(B_i P_{M,2}^T)\right) B_n P_2 \tilde{I}_M$$

where

$$P_2 = \overbrace{P_{M,2}R_M \ldots P_{M,2}R_M}^{n-1 \text{ times}} \tag{10}$$

and other notations are the same as for the previous algorithm. The flow graph of this algorithm is shown in Fig.2.

4 ELT ARCHITECTURE

For efficient implementation of ELT we propose the combined pipelined-perfect shuffle network (see Fig.4). Pipeline part consists of K arithmetic units connected in chain and every arithmetic unit is responsible for sequential calculation of $M/2$ butterflies per block. This is possible because in the polyphase structure of Fig.1 the data are processed block by block, and every block is formed in M stages as M input data sequentially arrive. The processing of data in pipeline part is performed with the same rate as that of input data.

Because of structure of butterfly matrices D_i the delay-invert (DI) memory unit is used to arrange pairs of input data for butterflies sequentially in time (see Fig.3c). DI unit inverts with $M/2$ delay the sequence $x_0, x_1, \ldots, x_{M/2-1}$ to $x_{M/2-1}, \ldots, , x_1, x_0$ for every input

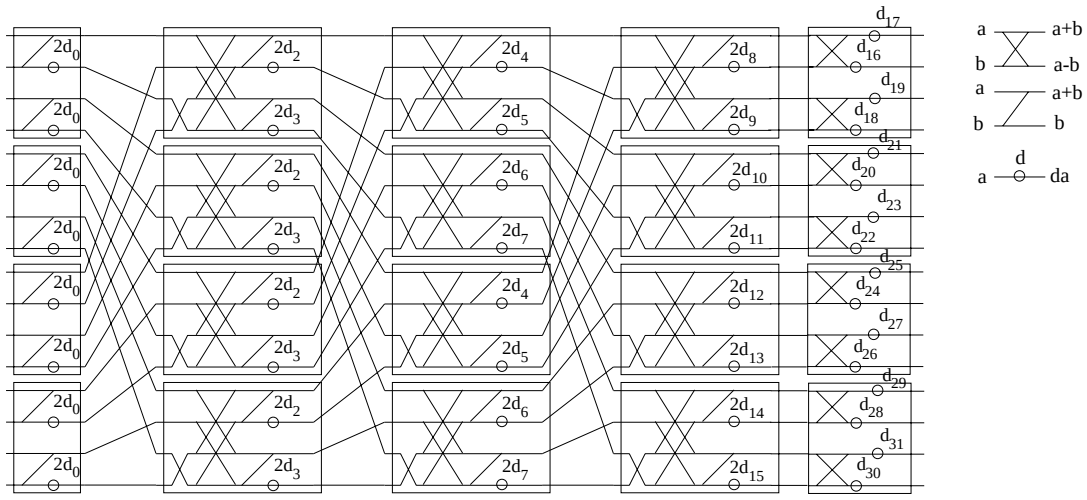

Figure 2: Constant geometry DCT-IV algorithm derived from Wang's DST-IV algorithm [6]

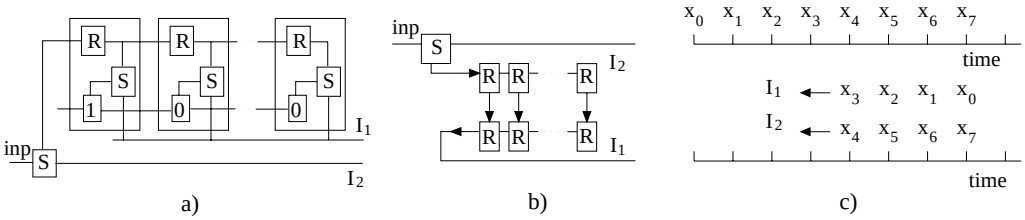

Figure 3: Delay-invert (DI) realizations and DI example for $M = 8$

block. Two realizations of DI are presented in Fig.3a-3b. The first is composed of the $M/2$ shift registers and switches. First $M/2$ stages $x_0, x_1, \ldots, x_{M/2-1}$ data of a block are shifted to the $M/2$ registers. Next $M/2$ stages due to the shifted one bit signal inverted data occur on the output.

Because of processing at input data rate and pipelining, i.e., compressing M channels in polyphase structure to two in our case, the delays z^{-1}, z^{-2}, distributed among channels and represented by matrices Z_i in factorizations (3),(4), will be provided by the common delays $z^{-M}, z^{-M/2}$, implemented as blocks of shift registers (BSR) as in Fig.4.

In our structure arithmetic units (PE) are performing calculations $M/2$ stages staying idle the other $M/2$. This can be avoided by two-channel processing. In this case DI unit in Fig.3b can be used and the common delays will be of length z^{-2M} and z^{-M}. In our case, because the next batch of input data pairs is delayed by $M/2$ stages, then the number of shift registers can be set to $z^{-M}, z^{-M/2}$ performing shifting by SRB only for those cycles when PE from the left is loaded.

Then data are collected for computation of DCT-IV with perfect-shuffle network (PSN). This is performed by the column of $M/4$ processor elements (PE). Every PE has 4 inputs and 4 outputs. The two outputs of p-th PE, $p = 0, 1, \ldots, M/4-1$, are connected to two outputs of $\lfloor \frac{2p \bmod M}{2} \rfloor$ PE, and the other two to the inputs of $\lfloor \frac{2p+1 \bmod M}{2} \rfloor$ PE. In other words, indexing the inputs and outputs of PE's column as $0, 1, \ldots, M-1$, the k-th output is connected to $2k \bmod M$ input.

Analysing the presented CG DCT-IV algorithms (see also Fig.2), one can see that in every stage of algorithm (index i) 4 point operations are followed by fixed reordering $P^T_{M,2}$ (perfect shuffle). The network connecting PEs serves for reordering $P^T_{M,2}$ and PE column sequentially calculates multiplication by matrices B_i. Comparing the two algorithms in Section 3, one can see that the block diagonal matrices B_{n-i} in the first one contain two equal coefficients in a block, while the other has two different. The choice of the first one in our implementation is aimed at the reduction of memory to hold these coefficients.

Every PE contains two multipliers. The changing of signs by $\tilde{I}$ is performed at the end of pipelined stage. The different operations performed by PE on the input $\vec{x} = (x_1, x_2, x_3, x_4)$ are coded as $O_1 = O(\vec{x}, d, *, 000)$, $O_2 = O(\vec{x}, d, *, 001)$, $O_3 = O(\vec{x}, d_1, d_2, 010)$, $O_4 = O(\vec{x}, d_1, d_2, 011)$, $O_5 = O(\vec{x}, d_1, d_2, 100)$.

$$O_1 : M_2 \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 0 & d \end{pmatrix} R\vec{x};\ O_2 : M_2 \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & d & 0 \\ 0 & 0 & 0 & d \end{pmatrix} \vec{x};$$

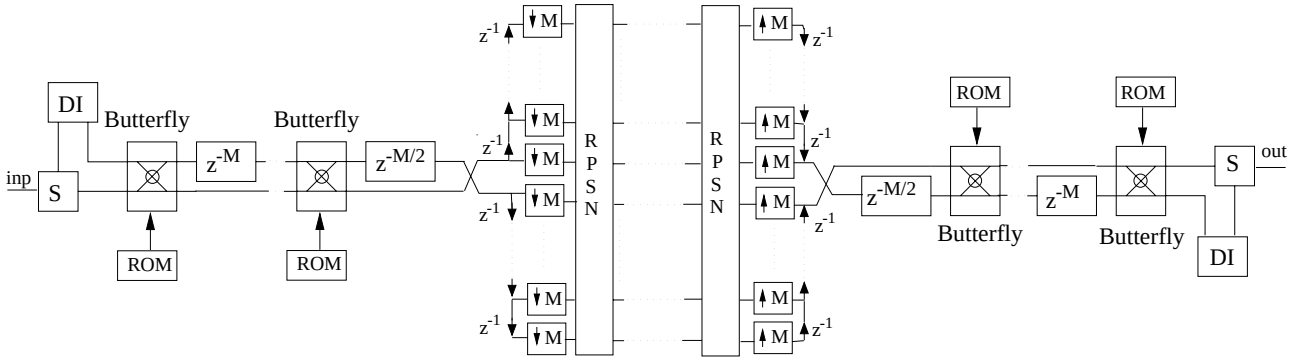

Figure 4: ELT architecture, analysis and synthesis parts

$$O_3 : M_1 \begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & d_1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & d_2 \end{pmatrix} \vec{x};\ O_4 : \begin{pmatrix} d_1 & 0 & 0 & 0 \\ 0 & d_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \vec{x},$$

$$O_5 : \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & d_1 & 0 \\ 0 & 0 & 0 & d_2 \end{pmatrix} \vec{x},$$

The DCT-IV calculation is performed by the algorithm

parallel input $x_0, \ldots, x_{M-1}$
Reordering P_1
For $p = 0$ **to** $M/4 - 1$ **do in parallel**
  $O(\vec{x}, 2d_0^0, *, 001)$
end p
For $i = 1$ **to** $n - 2$ **do in sequential**
  Perf.shuffle $P^T_{M,2}$ **in parallel**
  For $p = 0$ **to** $M/4 - 1$ **do in parallel**
    $k = p \bmod 2^i$, $O(\vec{x}, 2d_k^i, *, 001)$
  end p
end i
For $p = 0$ **to** $M/4 - 1$ **do in parallel**
  $O(\vec{x}, 2d_{n-1}^{2p}, 2d_{n-1}^{2p+1}, 010)$
  ***)** $O(\vec{x}, 2d_n^{4p}, 2d_n^{4p+1}, 011)$, $O(\vec{x}, 2d_n^{4p+2}, 2d_n^{4p+3}, 100)$
end p
end

Operations $O_1 - O_3$ differ only by exchanging of intermediate results and PE unit performs one of them by changing appropriately the states of switches.

Operations *) are not necessary if scaled outputs could be processed in subbands. For synthesis part the transpose structure is used.

The $\log M$ iterations of PSN network [10], with $M/4$ processor units for our case when performed at the input data rate do not influence the real-time processing because of order M decimation. Decimators are synchronized with the input switch to correctly form data blocks for DCT-IV module.

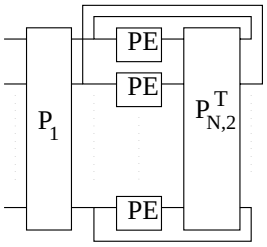

Figure 5: Perfect shuffle network

References

[1] H.S. Malvar, Signal Processing with Lapped Transforms. Dedham, MA: Artech House, 1992.

[2] H.S. Malvar, "Modulated QMF filter banks with perfect reconstruction," Electron. Lett., vol. 26, no.13, June 1990, pp. 906-907.

[3] J.Princen, A.Johnson, and A.Bradley, "Sub-band transform coding using filterbank designs based on time-domain aliasing cancelation," Proc. ICASSP, pp.2161-2164, 1987.

[4] R.L. Queroz and K.R. Rao, "Extended lapped transform in image coding," IEEE Trans. on Image Processing, vol. 4, no. 6, June 1995.

[5] L.F. Vargas and H.S. Malvar, "ELT-based Wavelet coding of high-fidelity audio signals," Proc. of ISCAS'1993, vol.1, pp. 124-127.

[6] Z. Wang. "Fast discrete sine transform algorithms," Signal Processing 19 (1990) 91-102

[7] J. Astola and D. Akopian, "Architecture-oriented regular algorithms for discrete sine and cosine transforms," IS & T/ SPIE's Symp. on Electronic Imaging, Image and Video Proc.-IV, Jan. 1996.

[8] J.Cooley and J. Tukey, "An algorithm for the machine calculation of the complex Fourier series", Math. Comput., vol.19, pp. 297-301, Apr. 1965.

[9] M.C. Pease, "An adaptation of the fast Fourier transform for parallel processing," J. ACM, vol. 15, pp. 252-264, 1968.

[10] H.S. Stone, "Parallel processing with the perfect shuffle," IEEE Trans. Comput. C-20, pp.153-161, 1971.