ON EFFICIENT IMPLEMENTATION OF MULTIDIMENSIONAL MULTIRATE FILTERS DERIVED FROM ONE-DIMENSIONAL FILTERS

Tsuhan Chen
AT&T Research
Room 4C528, 101 Crawfords Corner Road, Holmdel, NJ 07733, USA
Tel: +1 908 949-2708 Fax: +1 908 957-8388
e-mail: tsuhan@research.att.com

ABSTRACT

We study the efficient implementation of multidimensional (MD) filters used in multirate systems. These filters, typically having parallelepiped-shaped passband supports, can be derived from one-dimensional (1D) prototype filters. The resulting nonseparable MD filters have separable polyphase components that are combinations of the polyphase components of the 1D prototypes, so efficient implementation exists. We show that, for the two-dimensional case, all the polyphase components of the 1D prototypes are utilized. Therefore, there is no design overhead in this scheme.

1 INTRODUCTION

Multidimensional (MD) filters used in multirate applications, e.g., decimation filters, interpolation filters, and analysis/synthesis filter banks, typically have parallelepiped-shaped passband supports. In [1], we have shown that these filters can be designed by starting from one-dimensional (1D) prototype filters.

In this paper, we will show that nonseparable MD filters designed with the method in [1] actually have separable polyphase components that are combinations of the polyphase components of the 1D prototypes, so efficient implementation exists. We will present the condition under which all the polyphase components of the 1D prototypes are utilized, so that no design overhead is introduced. We will show that, for the two-dimensional (2D) case, this condition is always satisfied by our design method.

Notations: In this paper, boldfaced letters denote matrices and column vectors. The notations $\mathbf{A}^T$, $\mathbf{A}^{-1}$ and $\mathbf{A}^{-T}$ denote the transpose, the inverse and the inverse transpose of the matrix $\mathbf{A}$, respectively. The i-th element of a vector $\mathbf{x}$ is denoted as either x_i or $[\mathbf{x}]_i$. The $D \times 1$ integer vector $\mathbf{n}$ is the 'space' domain index of D-dimensional discrete-time signals. The symbol $\mathcal{N}$ denotes the set of all $D \times 1$ integer vectors so that $\mathbf{n} \in \mathcal{N}$. The Fourier transform of $x(\mathbf{n})$ is given by $X(\boldsymbol{w}) = \sum_{\mathbf{n}\in\mathcal{N}} x(\mathbf{n}) e^{-j\boldsymbol{w}^T\mathbf{n}}$, where the $D \times 1$ real vector $\boldsymbol{w}$ is the frequency domain variable. The $\mathbf{M}$-fold decimated version of $x(\mathbf{n})$ is defined as $y(\mathbf{n}) = x(\mathbf{Mn})$. In the frequency domain, the relation is

$$Y(\boldsymbol{w}) = \frac{1}{J_{\mathbf{M}}} \sum_{\mathbf{k}\,\in\,\mathcal{N}(\mathbf{M}^T)} X\big(\mathbf{M}^{-T}(\boldsymbol{w} - 2\pi\mathbf{k})\big), \quad (1)$$

where $\mathcal{N}(\mathbf{M}^T)$ is the set of all integer vectors of the form $\mathbf{M}^T\mathbf{x}$, $\mathbf{x} \in [0,1)^D$. Also, $J_{\mathbf{M}}$ denotes the absolute determinant of $\mathbf{M}$, $|\det \mathbf{M}|$, which is equal to the number of elements in $\mathcal{N}(\mathbf{M}^T)$ or $\mathcal{N}(\mathbf{M})$. The $\mathbf{M}$-fold expanded version of $x(\mathbf{n})$ is defined as

$$y(\mathbf{n}) = \begin{cases} x(\mathbf{M}^{-1}\mathbf{n}) & \mathbf{n} \in LAT(\mathbf{M}) \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In the above equation, $LAT(\mathbf{M})$ (the *lattice* generated by $\mathbf{M}$) is the set of all vectors of the form $\mathbf{Mm}$, where $\mathbf{m} \in \mathcal{N}$. The frequency domain relation of expansion is

$$Y(\boldsymbol{w}) = X(\mathbf{M}^T\boldsymbol{w}). \quad (3)$$

To avoid aliasing in $\mathbf{M}$-fold decimation or to remove image components in $\mathbf{M}$-fold expansion, a decimation/expansion filter having passband support in $SPD(\pi\mathbf{M}^{-T}) \triangleq \{\pi\mathbf{M}^{-T}\mathbf{x} \,|\, \mathbf{x} \in [-1,1)^D\}$ is typically used. For other notations and preliminaries of MD multirate systems, please see [2, 3].

2 DESIGN METHOD

Suppose we need a decimation filter $h(\mathbf{n})$ for $\mathbf{M}$-fold decimation. A typical passband support for this is $SPD(\pi\mathbf{M}^{-T})$. In [1], it was shown that $h(\mathbf{n})$ can be designed by following these steps:

1. Decompose $\mathbf{M}$ into $\mathbf{M} = \mathbf{Q}\boldsymbol{\Lambda}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix with diagonal elements $\lambda_i > 0$, and λ_i is the greatest common divisor of the elements in the i-th column of $\mathbf{M}$. For example:

$$\mathbf{M} = \begin{bmatrix} 1 & 2 \\ -1 & 2 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}}_{\mathbf{Q}} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}}_{\boldsymbol{\Lambda}} \quad (4)$$

2. Using any 1D filter design algorithm, design 1D prototype filters $p_0(n), p_1(n), \ldots, p_{D-1}(n)$. Each filter $p_i(n)$ has the passband region in $-\frac{\pi}{J_{\mathbf{Q}}\lambda_i} \le \omega < \frac{\pi}{J_{\mathbf{Q}}\lambda_i}$.

3. Construct $h^{(s)}(\mathbf{n}) \triangleq p_0(n_0)p_1(n_1)\cdots p_{D-1}(n_{D-1})$.

4. Construct $h(\mathbf{n}) \triangleq c_0 h^{(s)}(\widehat{\mathbf{Q}}\mathbf{n})$, where $\widehat{\mathbf{Q}} \triangleq J_{\mathbf{Q}}\,\mathbf{Q}^{-1}$ and $c_0 = J_{\widehat{\mathbf{Q}}}$.

In [1], it was shown that the resulting passband support of $h(\mathbf{n})$ is indeed $SPD(\pi\mathbf{M}^{-T})$. Also, the passband and stopband ripples of $h(\mathbf{n})$ are bounded by certain functions of the passband and stopband ripples of prototype filters $p_i(n)$. Furthermore, many properties of the prototype filters, such as linear phase, stability and Nyquist property, are retained in $h(\mathbf{n})$.

It turns out that a filter designed using the above method actually has separable polyphase components, although the filter itself is not separable. This leads to efficient implementation of MD filters in which the complexity is reduced from the typical $\mathcal{O}(N^D)$ to $\mathcal{O}(N)$. In [1], this was shown only for the 2D case, and only for the case $\mathbf{Q} = \mathbf{M}$. The main purpose of this paper is to extend the results in [1] to the most general case.

3 EFFICIENT IMPLEMENTATION

Given a filter $h(\mathbf{n})$ designed as above, we now show that it has separable polyphase components. With respect of $\mathbf{M}$, the polyphase components of $h(\mathbf{n})$ are defined as follows (slightly different from those in [2, 3]):

$$e_{\mathbf{k}}(\mathbf{n}) \triangleq \begin{cases} h(\mathbf{n}) & \mathbf{n} = \mathbf{M}\mathbf{m} + \mathbf{k} \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where $\mathbf{k} \in \mathcal{N}(\mathbf{M})$, so totally there are $J_{\mathbf{M}}$ polyphase components. For the 1D case, this reduces to

$$e_k(n) \triangleq \begin{cases} h(n) & n = Mm + k \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

for $0 \le k < M$. Note that $h(\mathbf{n}) = \sum_{\mathbf{k}\in\mathcal{N}(\mathbf{M})} e_{\mathbf{k}}(\mathbf{n})$. Now, we can rewrite each $e_{\mathbf{k}}(\mathbf{n})$ as

$$\begin{aligned} e_{\mathbf{k}}(\mathbf{n}) &= \begin{cases} c_0 h^{(s)}(\widehat{\mathbf{Q}}\mathbf{n}) & \mathbf{n} = \mathbf{M}\mathbf{m} + \mathbf{k} \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} c_0 h^{(s)}(J_{\mathbf{Q}}\mathbf{\Lambda}\mathbf{m} + \widehat{\mathbf{Q}}\mathbf{k}) & \mathbf{n} = \mathbf{M}\mathbf{m} + \mathbf{k} \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} c_0 \displaystyle\prod_{i=0}^{D-1} p_i(J_{\mathbf{Q}}\lambda_i m_i + [\widehat{\mathbf{Q}}\mathbf{k}]_i) & \mathbf{n} = \mathbf{M}\mathbf{m} + \mathbf{k} \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{7}$$

Because $\mathbf{k} \in \mathcal{N}(\mathbf{M})$, we have $\widehat{\mathbf{Q}}\mathbf{k} \in \mathcal{N}(\widehat{\mathbf{Q}}\mathbf{M}) = \mathcal{N}(J_{\mathbf{Q}}\mathbf{\Lambda})$. This implies that $0 \le [\widehat{\mathbf{Q}}\mathbf{k}]_i < J_{\mathbf{Q}}\lambda_i$. Let $r_i(\mathbf{k}) \triangleq [\widehat{\mathbf{Q}}\mathbf{k}]_i$. We see that each term $p_i(J_{\mathbf{Q}}\lambda_i m_i + [\widehat{\mathbf{Q}}\mathbf{k}]_i)$ in (7) is indeed equal to the $J_{\mathbf{Q}}\lambda_i$-fold polyphase component of $p_i(n)$, i.e., $p_{i,r_i(\mathbf{k})}(n)$. Hence, we see that $h(\mathbf{n})$ can be implemented as the sum of separable polyphase components as shown in Figure 1. In this figure, J denotes $J_{\mathbf{M}}$, and $\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_J$ denote the vectors in $\mathcal{N}(\mathbf{M})$. There are $J_{\mathbf{M}}$ channels in this implementation and each channel is formed by D stages.

4 USE OF POLYPHASE COMPONENTS

An interesting issue worth some investigation is how the polyphase components of $p_i(n)$ are permuted in the implementation in Figure 1. In particular, we want to study whether all the polyphase components of the prototype filter $p_i(n)$ are utilized in the structure of Figure 1, which defines the *efficiency* of the proposed design method. In this section, we will show that this depends on whether the elements in each row of $\widehat{\mathbf{Q}}$ have a common factor. We shall call a matrix "rowwise prime" if the elements in each row of $\widehat{\mathbf{Q}}$ do not have a non-unity common factor.

Theorem 1 *If $\widehat{\mathbf{Q}}$ is rowwise prime, all the polyphase components of the 1D prototype filters are utilized in Figure 1.*

Proof: We need to prove that, for each dimension i, there exists $\mathbf{k} \in \mathcal{N}(\mathbf{M})$ such that $[\widehat{\mathbf{Q}}\mathbf{k}]_i = r_i$ for any $r_i \in [0, J_{\mathbf{Q}}\lambda_i)$. Denote the i-th row of $\widehat{\mathbf{Q}}$ as $\mathbf{q}_i^T$. Since $\widehat{\mathbf{Q}}$ is rowwise prime, so there is no common factor among elements of $\mathbf{q}_i^T$. Based on extended Euclid's theorem [4, Sec. 2.3], there exist $\mathbf{j}$ such that $\mathbf{q}_i^T\mathbf{j} = [\widehat{\mathbf{Q}}\mathbf{j}]_i = r_i$ for any integer r_i. Now, note that any integer vector $\mathbf{j}$ can written as $\mathbf{j} = \mathbf{M}\mathbf{m} + \mathbf{k}$ for some $\mathbf{k} \in \mathcal{N}(\mathbf{M})$. Therefore, we have

$$\begin{aligned} r_i &= [\widehat{\mathbf{Q}}(\mathbf{M}\mathbf{m} + \mathbf{k})]_i \\ &= [J_{\mathbf{Q}}\mathbf{\Lambda}\mathbf{m} + \widehat{\mathbf{Q}}\mathbf{k}]_i \\ &= J_{\mathbf{Q}}\lambda_i m_i + [\widehat{\mathbf{Q}}\mathbf{k}]_i \end{aligned} \tag{8}$$

Because $0 \le r_i < J_{\mathbf{Q}}\lambda_i$ and $0 \le [\widehat{\mathbf{Q}}\mathbf{k}]_i < J_{\mathbf{Q}}\lambda_i$, we can conclude that $m_i = 0$, and $r_i = [\widehat{\mathbf{Q}}\mathbf{k}]_i$. Hence, we have proved the theorem. △△△

Furthermore, the following corollaries can be easily verified:

Corollary 1.1 *If elements of the i-th row of $\widehat{\mathbf{Q}}$ has a common factor R, then not all the polyphase components of the prototype filter $p_i(n)$ are used in Figure 1. More specifically, only the 0-th, R-th, $2R$-th,... polyphase components are used.*

Corollary 1.2 *In each stage i of Figure 1, the polyphase components $p_{i,r_i(\mathbf{k})}(m_i)$ that are included, are used for the same number of times.*

5 THE TWO-DIMENSIONAL CASE

In this section, we will show that, for the 2D case, our design method always guarantees that *all* polyphase components of the prototype filters $p_i(n)$ are used in Figure 1. This means that there is no overhead when we derive a 2D filter from 1D prototypes using our method. In order to prove this statement, we start with the following theorem:

Theorem 2 *If* $\mathbf{Q}$ *is not columnwise prime,* $\widehat{\mathbf{Q}}$ *is not rowwise prime.*

Proof: If $\mathbf{Q}$ is not columnwise prime, then

$$\mathbf{Q} = \mathbf{P} \underbrace{\begin{bmatrix} \lambda_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{D-1} \end{bmatrix}}_{\mathbf{\Lambda}} \tag{9}$$

where λ_i are not all unity. It is easily verified that

$$\widehat{\mathbf{Q}} = \underbrace{\begin{bmatrix} \lambda_1\lambda_2\cdots\lambda_{D-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_0\lambda_1\cdots\lambda_{D-2} \end{bmatrix}}_{\widehat{\mathbf{\Lambda}}} \widehat{\mathbf{P}} \tag{10}$$

Since $\widehat{\mathbf{\Lambda}}$ is not an identity matrix, $\widehat{\mathbf{Q}}$ is not rowwise prime. △△△

Corollary 2.1 *If* $\mathbf{Q}$ *is not rowwise prime,* $\widehat{\mathbf{Q}}$ *is not columnwise prime.*

Note that both Theorem 2 and Corollary 2.1 are true for any number of dimensions. Now we are ready to prove the following theorem that is true only for the 2D case.

Theorem 3 *For the 2D case,* $\mathbf{Q}$ *is columnwise prime if and only if* $\widehat{\mathbf{Q}}$ *is rowwise prime.*

Proof: It is easily verified that $\widehat{\widehat{\mathbf{P}}} = J_{\mathbf{P}}^{D-2}\mathbf{P}$. So, for the 2D case, $\widehat{\widehat{\mathbf{P}}} = \mathbf{P}$. From Theorem 2, we know that if $\mathbf{Q}$ is not columnwise prime, $\widehat{\mathbf{Q}}$ is not rowwise prime. Replacing $\mathbf{Q}$ with $\widehat{\mathbf{Q}}$ in Corollary 2.1, we obtain that if $\widehat{\mathbf{Q}}$ is not rowwise prime, then $\widehat{\widehat{\mathbf{Q}}}$ is not columnwise prime, i.e., $\mathbf{Q}$ is not columnwise prime. Hence, we have proved that $\mathbf{Q}$ is columnwise prime if and only if $\widehat{\mathbf{Q}}$ is rowwise prime. △△△

In the design method described in Section 2, all the common factors in columns of $\mathbf{M}$ are absorbed into by $\mathbf{\Lambda}$, so $\mathbf{Q}$ is columnwise prime. For the 2D case, this implies that $\widehat{\mathbf{Q}}$ is rowwise prime. With Theorem 1, we therefore conclude that all polyphase components of the prototype filters are utilized in the implementation, so the proposed method is indeed efficient in that no design overhead is involved.

6 EXAMPLES

We have proved that the efficiency of the design method described in Section 2 is guaranteed for the 2D case. However, for the three-dimensional (3D) case or higher dimensions, this is not always true. In this section, we present a 2D example and a 3D example to illustrate these.

Consider the following columnwise prime matrix that defines the hexagonal decimation [3]

$$\mathbf{M} = \begin{bmatrix} 1 & 1 \\ -2 & 2 \end{bmatrix} \tag{11}$$

In this case, $\mathbf{Q} = \mathbf{M}$ and

$$\widehat{\mathbf{Q}} = \begin{bmatrix} 2 & -1 \\ 2 & 1 \end{bmatrix} \tag{12}$$

which is indeed rowwise prime and verifies Theorem 3. Suppose we build a 2D filter with passband support $SPD(\pi\mathbf{M}^{-T})$ using the proposed design method and implementation. Since $J_{\mathbf{M}} = 4$, there are four channels in this implementation, and each channel has two stages. Examining $[\widehat{\mathbf{Q}}\mathbf{k}]_i$ for

$$\mathbf{k} \in \mathcal{N}(\mathbf{M}) = \left\{ \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\} \tag{13}$$

we get

$$\widehat{\mathbf{Q}}\mathbf{k} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \end{bmatrix}, \text{ or } \begin{bmatrix} 1 \\ 3 \end{bmatrix} \tag{14}$$

Note that both $[\widehat{\mathbf{Q}}\mathbf{k}]_0$ and $[\widehat{\mathbf{Q}}\mathbf{k}]_1$ cover all elements in $\{0, 1, 2, 3\}$. Therefore, all the four polyphase components of $p_0(n)$ and the four of $p_1(n)$ appear in the implementation of Figure 1. In other words, no design overhead is created by the proposed method.

For the 3D case, consider the following columnwise prime matrix

$$\mathbf{M} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 0 & 1 \end{bmatrix} \tag{15}$$

In this case, $\mathbf{Q} = \mathbf{M}$, but

$$\widehat{\mathbf{Q}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ -3 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -3 & 0 & 2 \end{bmatrix} \tag{16}$$

which is not rowwise prime. Recall that this never happens in the 2D case. We follow the design method and implement $h(\mathbf{n})$ as in Figure 1. Since $J_{\mathbf{M}} = 2$, there are two channels in this implementation, and each channel has three stages. Let us examine $[\widehat{\mathbf{Q}}\mathbf{k}]_i$ for

$$\mathbf{k} \in \mathcal{N}(\mathbf{M}) = \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \right\} \tag{17}$$

We have

$$\widehat{\mathbf{Q}}\mathbf{k} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \text{ or } \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \tag{18}$$

Note that $[\widehat{\mathbf{Q}}\mathbf{k}]_1$ never equals to one. We conclude that the "first" polyphase component of $p_1(n)$, $p_{1,1}(m_1)$, is not used at all in the implementation of Figure 1. Instead, its 0-th polyphase component, $p_{1,0}(m_1)$, is used in both channels.

7 CONCLUDING REMARKS

In this paper, we studied the efficient implementation of MD filters designed for multirate applications. We showed that these filters can be derived from 1D prototype filters. The resulting MD filters have separable polyphase components that are combinations of the polyphase components of the 1D prototypes, so they can be implemented efficiently. For the 2D case, we showed that all the polyphase components the 1D prototypes are utilized in our implementation, so there is no design overhead.

Here we make a further remark on the efficient implementation. In [1], it was shown that we can guarantee the phase linearity of $h(\mathbf{n})$ by starting from linear-phase 1D prototype filters. In this case, the polyphase components in Figure 1 has certain symmetry that can be exploited further to improve the implementation efficiency. The details are similar to those in [3, Figure 2.4-3 and Problem 4.17].

We conclude the paper by describing some open problems that remain to be solved:

1. In [1], it was shown that a filter having passband support $SPD(\pi\mathbf{H}^{-T})$ where $\mathbf{H}$ is a matrix with rational elements can also be derived from 1D prototypes. How do the results in this paper extend to the case of rational $\mathbf{H}$?

2. Our design method does not guarantee design efficiency for 3D or higher dimensions. That is, design overhead may exist for certain $\mathbf{M}$. How can one modify the design method to avoid that?

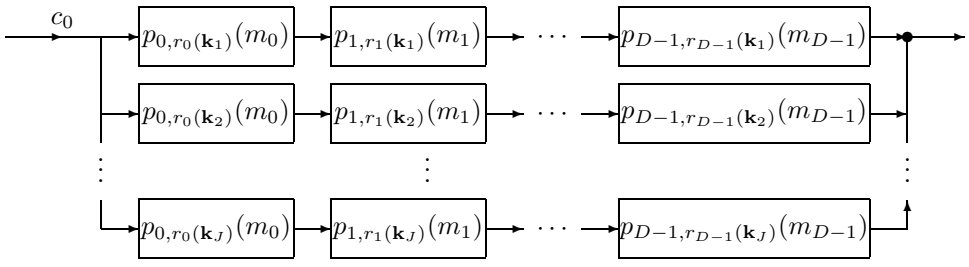

Figure 1: Efficient implementation of $h(\mathbf{n})$.

References

[1] T. Chen and P. P. Vaidyanathan, "Multidimensional multirate filters and filter banks derived from one dimensional filters," *IEEE Trans. on Signal Processing*, vol. 41, no. 5, pp. 1749–1765, May 1993.

[2] E. Viscito, and J. P. Allebach, "The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices," *IEEE Trans. on CAS*, vol. 38, no. 1, pp. 29–41, January 1991.

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

[4] N. K. Bose, *Digital Filters*, Elsevier Science Publ., 1985.