A Novel Implementation Method for Anticausal IIR Digital Filter Banks

Umut Sezen[†]
Department of Electrical and Electronic Engineering
Hacettepe University

***Abstract* — A novel and efficient method which addresses both the perfect reconstruction and the signal extension problems of polyphase IIR filter banks is presented. The method involves embedding the final filter states of the analysis filters into the analysis output in a nonexpansive way in the analysis stage and still satisfying perfect reconstruction by solving a system of linear equations in the synthesis stage in order to recover the lost data due to filter state embedding. The superiority of the proposed method is justified by the presented image coding results.**

1 Introduction

Perfect reconstruction property of a maximally decimated infinite impulse response (IIR) filter bank based on a polyphase implementation as shown in Figure 1 requires the synthesis polyphase filters to be the inverses of the analysis polyphase filters. In most cases the inverses have poles outside the unit circle resulting in unstable synthesis filters. This difficulty is overcome by implementing the inverses as anticausal filters because anticausal filters with poles outside the unit circle are stable. Anticausal inverses for IIR filter banks were introduced by Husoy and Ramstad [1], and then conditions for perfect reconstruction were investigated by Chen and Vaidyanathan [2]. Thus perfect reconstruction is achieved by decomposing polyphase filters into minimum- and maximum-phase components in the analysis stage, e.g. $E(z) = B(z)G(z)$, and implementing the anticausal inverse of the maximum-phase component and the causal inverse of the minimum-phase component in the synthesis stage, e.g. $R(z) = G^{-1}(z^{-1})B^{-1}(z)$, provided that the initial filter states of the anticausal inverse are set correctly related to the final filter states of the maximum-phase component [3, 4].

In this paper we suggest a new method to include the final filter states of the analysis maximum-phase filters into their filter output in a nonexpansive way and recovering the lost information due to nonexpansive embedding by solving a system of linear equations which is simply described by a constant matrix. The method improves on the previous methods [4, 5] to a more powerful and more complete (or comprehensive) solution covering the signal extension solutions at the boundaries reducing the transient effects [6, 7]. We will start with explaining anticausal inverses and the relationship between the initial filter states of the anticausal inverse and the final filter states of the maximum-phase filter. Later we will explain our embedded filter states (EFS) method in terms of the procedures to be implemented in both the analysis and synthesis stages. Finally we will present the image compression results and our conclusion.

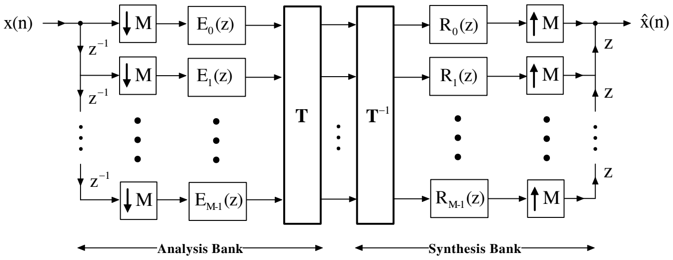


Figure 1: Polyphase implementation of a uniform maximally decimated M-channel filter bank where $\mathbf{T}$ is a modulation matrix e.g. the DFT matrix.

2 Anticausal inverses

Given that a causal stable maximum-phase filter has following state-space description

$$\begin{bmatrix}\mathbf{s}(n+1)\\ y(n)\end{bmatrix} = \underbrace{\begin{bmatrix}\mathbf{A} & \mathbf{B}\\ \mathbf{C} & \mathbf{D}\end{bmatrix}}_{\mathbf{R}}\begin{bmatrix}\mathbf{s}(n)\\ x(n)\end{bmatrix} \qquad (1)$$

where $\mathbf{s}(n) = \begin{bmatrix}s_1(n) & s_2(n) & \cdots & s_N(n)\end{bmatrix}^T$ is the state-vector, $x(n)$ is the filter input, $y(n)$ is the filter output, and $\mathbf{R}$ is said to be the realisation matrix of the implementation. After processing an L-sample filter input $x(n)$, we would have an L-sample filter output $y(n)$, and the final state-vector $\mathbf{s}(L)$.

Let us to recover the filter input $x(n)$ from the filter output $x(n)$ using the inverse filter $G^{-1}(z)$. As the inverse filter $G^{-1}(z)$ will be unstable, we have to implement the inverse filter as an anticausal filter $\hat{G}(z) = G^{-1}(z^{-1})$. Let us write down the state-space description for the anticausal inverse filter

[†] E-mail: u.sezen@ee.hacettepe.edu.tr

$\hat{G}(z)$,

$$\begin{bmatrix}\hat{\mathbf{s}}(n+1)\\ \hat{y}(n)\end{bmatrix} = \underbrace{\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{B}}\\ \hat{\mathbf{C}} & \hat{\mathbf{D}}\end{bmatrix}}_{\hat{\mathbf{R}}} \begin{bmatrix}\hat{\mathbf{s}}(n)\\ \hat{x}(n)\end{bmatrix} \tag{2}$$

and let the realisation matrix of $\hat{G}(z)$ to be the inverse of the realisation matrix of $G(z)$, i.e. $\hat{\mathbf{R}} = \mathbf{R}^{-1}$.

Since the filter input of $\hat{G}(z)$ is the time-reversed filter output $G(z)$ of i.e. $\hat{x}(n) = y(L-1-n)$, the filter output of $\hat{G}(z)$ should also be the time-reversed filter input of $G(z)$ i.e.

$$\hat{y}(n) = x(L-1-n), \tag{3}$$

when we let the initial state-vector of $\hat{G}(z)$ be the final state-vector of $G(z)$ i.e. $\hat{\mathbf{s}}(0) = \mathbf{s}(L)$ [2]. Ultimately the two state-vectors are related as

$$\hat{\mathbf{s}}(n) = \mathbf{s}(L-n) \tag{4}$$

which can be derived using equations (2) and (2) [2].

3 Embedded filter states (EFS) method

The embedded filter states (EFS) method can be explained simply that in the analysis stage, the final filter states of $G(z)$ are embedded into the filter output in a nonexpansive way in order to ensure that the final output has the same number of samples as the filter input. In other words after processing L-sample input and appending the final filter states $\mathbf{s}(L)$ (N samples) to the filter output, only the last L samples are retained. In the synthesis stage the procedure is to recover the input signal of the analysis stage using the anticausal inverses of the filters used in the analysis stage. The following sections explain the use of linear algebra concepts in order to recover the first N-samples of the input signal although the first N-samples of the filter outputs were discarded and how to incorporate the signal extension concepts within the EFS algorithm.

4 Final-state and final-output equations

Using the iterative next-state equations given in (1), we can arrive at the non-iterative representation of final filter state-vector $\mathbf{s}(N)$, in terms of an filter N-sample input-vector $\mathbf{x} = \begin{bmatrix}x(0) & x(1) & \cdots & x(N-1)\end{bmatrix}^T$ and initial filter state-vector $\mathbf{s}(0)$

$$\mathbf{s}(N) = \mathbf{Q}\,\mathbf{s}(0) + \mathbf{P}\,\mathbf{x} \tag{5}$$

where both $\mathbf{Q}$ and $\mathbf{P}$ are $N \times N$ matrices. Let us call this equation the *final-state equation* and the matrices $\mathbf{Q}$ and $\mathbf{P}$ as the *initial-state-to-final-state* (ISFS) matrix and the *input-to-final-state* (IFS) matrix respectively. Note that, for FIR filters $\mathbf{Q} = 0$.

Similarly we can write down the filter output-vector $\mathbf{y}$ in terms of filter input-vector $\mathbf{x}$ and initial filter state-vector $\mathbf{s}(0)$

$$\mathbf{y} = \mathbf{V}\,\mathbf{s}(0) + \mathbf{U}\,\mathbf{x} \tag{6}$$

where both $\mathbf{V}$ and $\mathbf{U}$ are $N \times N$ matrices. Let us call this equation the *final-output equation* and the matrices $\mathbf{V}$ and $\mathbf{U}$ as the *input-to-final-output* (IFO) matrix and the *initial-state-to-final-output* (ISFO) matrix respectively.

The input vector $\mathbf{x}$ can be recovered from the final-state equation by solving the system of linear equations represented by

$$\mathbf{P}\,\mathbf{x} = \mathbf{s}(N) - \mathbf{Q}\,\mathbf{s}(0) \tag{7}$$

with known $\mathbf{s}(N)$ and $\mathbf{s}(0)$. Thus the input vector $\mathbf{x}$ is given by

$$\mathbf{x} = \mathbf{P}^{-1}\left[\mathbf{s}(N) - \mathbf{Q}\,\mathbf{s}(0)\right]. \tag{8}$$

This equation reduces to

$$\mathbf{x} = \mathbf{P}^{-1}\mathbf{s}(N). \tag{9}$$

with zero initial conditions. This equation lets us to recover the input signal from the final states of the filter.

4.1 Derivation of IFS and IFO matrices: P and U

Let us define $\mathbf{i}_k$ as the k-th column of an $N \times N$ identity matrix $\mathbf{I}_N$, i.e.

$$\mathbf{I}_N = \begin{bmatrix}\mathbf{i}_1 & \mathbf{i}_2 & \cdots & \mathbf{i}_k & \cdots & \mathbf{i}_N\end{bmatrix}, \tag{10}$$

let $\mathbf{s}^{(k)}(n)$ and $\mathbf{y}^{(k)}$ be the final filter state-vectors and the filter output-vectors—containing the last N samples of the filter output—, obtained after processing the unit filter inputs $\mathbf{x}^{(k)} = \mathbf{i}_k$ and with zero initial conditions $\mathbf{s}^{(k)}(0) = 0$ using the state-space description given by (1), i.e.

$$\begin{array}{rcl}\mathbf{x}^{(k)} & = & \mathbf{i}_k\\ \mathbf{s}^{(k)}(0) & = & 0\end{array} \tag{11}$$

where $1 \le k \le N$. Then IFS matrix $\mathbf{P}$ is given by

$$\mathbf{P} = \begin{bmatrix}\mathbf{s}^{(1)}(N) & \cdots & \mathbf{s}^{(k)}(N) & \cdots & \mathbf{s}^{(N)}(N)\end{bmatrix}. \tag{12}$$

Hence each $\mathbf{s}^{(k)}(N)$ is the weight-vector of the input sample $x(k-1)$ on the final filter state-vector $\mathbf{s}(N)$.

Similarly IFO matrix is given by

$$\mathbf{U} = \begin{bmatrix}\mathbf{y}^{(1)} & \mathbf{y}^{(2)} & \cdots & \mathbf{y}^{(k)} & \cdots & \mathbf{y}^{(N)}\end{bmatrix}. \quad (13)$$

In most cases except the application of EFS method to infinite-length signals in fragments we will always have zero input conditions eliminating the need to derive ISFS and ISFO matrices $\mathbf{Q}$ and $\mathbf{V}$. However $\mathbf{Q}$ and $\mathbf{V}$ can be calculated in the same manner, where in this case input vectors are initialized to zero, $\mathbf{x}^k = 0$ and the input state vectors are unit vectors, $\mathbf{s}^k(0) = \mathbf{i}_k$.

4.2 Derivation of final-state and final-output equation matrices under signal extension

In real applications e.g. image coding, we have to consider signal extension in order to ensure border continuity. Here we will assume that the signal is extended by N samples at the left boundary, that is, filter inputs $\mathbf{x}^{(k)}(n)$ in the algorithms above in Section 4.1 would be $2N$ samples instead of N samples. For example, below we present the necessary filter input value used in Section 4.1 in order to obtain the correct IFS matrix $\mathbf{P}$ according to the half-sample symmetric signal extension:

Half-sample symmetry at the left-boundary (LHS)

$$\mathbf{x}^{(k)} = \begin{bmatrix}\mathbf{i}_{N-k+1}^T & \mathbf{i}_k^T\end{bmatrix}^T, \qquad 1 \le k \le N \quad (14)$$

Let $\mathbf{P}_L$ to represent the IFS matrix supporting signal extension at the left-boundary for future correspondence.

Although signal extensions at the left-boundary can be reflected exactly with the calculation of the IFS matrix $\mathbf{P}$, signal extension at the right-boundary can not be reflected exactly but only approximately. In order to do so, we are going to estimate the output samples, $y(L)$ to $y(L+N-1)$, which correspond to the filter output for the right-boundary extended samples, $x(L)$ to $x(L+N-1)$ in terms of the final filter state-vector $\mathbf{s}(L)$, i.e.

$$\mathbf{y}_R \cong \tilde{\mathbf{y}}_R = \mathbf{K}\mathbf{s}(L) \quad (15)$$

where $\mathbf{y}_R = \begin{bmatrix}y(L) & y(L+1) & \cdots & y(L+N-1)\end{bmatrix}^T$ and $\mathbf{K}$ is an $N \times N$ matrix.

The expression for $\mathbf{K}$ is derived by considering a N-sample input signal. Let $\mathbf{U}_R$ represent the IFO matrix derived with zero initial conditions using the filter input-vectors defined for an N-sample signal extension at the right boundary, e.g.

Half-sample symmetry at the right-boundary (RHS)

$$\mathbf{x}^{(k)} = \begin{bmatrix}\mathbf{i}_k^T & \mathbf{i}_{N-k+1}^T\end{bmatrix}^T, \qquad 1 \le k \le N \quad (16)$$

and $\mathbf{P}$ represent the IFS matrix derived with the unit input vectors as given in (11). Let us write down the following final-output and final-state equations for zero initial conditions

$$\begin{array}{rcl}\mathbf{y}_R & = & \mathbf{U}_R\,\mathbf{x} \\ \mathbf{s}(N) & = & \mathbf{P}\,\mathbf{x}\end{array} \quad (17)$$

and the relationship between the two quantities as

$$\mathbf{y}_R = \mathbf{K}\,\mathbf{s}(N) \quad (18)$$

then $\mathbf{K}$ is given by

$$\mathbf{K} = \mathbf{U}_R\,\mathbf{P}^{-1}. \quad (19)$$

Note that $\mathbf{K}$ is constant and needs to be initialized only once for each filter.

5 EFS algorithm

Let $G(z)$ be an N-th order maximum-phase filter, $x(n)$ denote the L-sample filter input, $y(n)$ denote the corresponding L-sample filter output, $v(n)$ denote the final L-sample EFS analysis output and $\hat{v}(n)$ denote the final L-sample EFS analysis output. The aim is to achieve $\hat{v}(n) = x(n)$.

Analysis Stage:

1. Initialize $G(z)$ with the filter-state vector $\mathbf{t}(0) = \mathbf{P}_L\,\mathbf{r}$ where $\mathbf{r}$ holds the first N samples, $x(0)$ to $x(N-1)$, of the input signal. Then filter the rest of the $(L-N)$ samples, $x(N)$ to $x(L-1)$, with $G(z)$, and obtain the filter output $z(n)$ and final state-vector $\mathbf{t}(L-N)$, where the filter output $z(n)$ is

$$z(n) = \{y(N), y(N+1), \ldots, y(L-1)\} \quad (20)$$

2. Derive the approximate filter output for the right-boundary extended samples using the final filter state-vector

$$\tilde{\mathbf{y}}_R = \mathbf{K}\,\mathbf{t}(L-N). \quad (21)$$

3. Obtain the EFS analysis output $v(n)$ by appending the calculated boundary samples to the filter output $z(n)$, i.e.

$$v(n) = \{y(N), \ldots, y(L), \tilde{y}_R(0), \ldots, \tilde{y}_R(N-1)\} \quad (22)$$

Synthesis Stage:

1. Initialize the anticausal inverse filter $\hat{G}(z)$ with the last N samples of $v(n)$ by solving the linear equation $\mathbf{K}\hat{\mathbf{t}}(0) = \tilde{\mathbf{y}}_R$, i.e.

$$\hat{\mathbf{t}}(0) = \mathbf{K}^{-1}\,\tilde{\mathbf{y}}_R. \quad (23)$$

Then set $\hat{u}(n)$ to be time-reversed version of the remaining samples of $v(n)$, i.e.

$$\hat{u}(n) = \{y(L), y(L-1), \ldots, y(N)\} \quad (24)$$

2. Filter $\hat{u}(n)$ with $\hat{G}(z)$, and obtain the filter output $\hat{z}(n)$ and final-state vector $\hat{\mathbf{t}}(L-N)$. From (3) and (4) we can deduce that $\hat{z}(n)$ and $\hat{\mathbf{t}}(L-N)$ would be equal to

$$\hat{z}(n) = \{x(L), x(L-1), \ldots, x(N)\} \qquad (25)$$

and

$$\hat{\mathbf{t}}(L-N) = \mathbf{t}(0). \qquad (26)$$

3. Solve the system of linear equations given by $\mathbf{P}_L\,\hat{\mathbf{r}} = \hat{\mathbf{t}}(L-N)$. The solution of the linear equation is given by

$$\hat{\mathbf{r}} = \mathbf{P}_L^{-1}\,\hat{\mathbf{t}}(L-N). \qquad (27)$$

From (9), (26) and (27) and we can deduce that $\hat{\mathbf{x}}$ would be equal to

$$\hat{\mathbf{r}} = \begin{bmatrix} x(0) & x(1) & \cdots & x(N-1) \end{bmatrix}^T \qquad (28)$$

4. Obtain the EFS synthesis output $\hat{v}(n)$ by appending the filter output $\hat{z}(n)$ in reverse order to the elements of the solution vector $\hat{\mathbf{r}}$, i.e.

$$\hat{v}(n) = \{\hat{r}_1, \ldots, \hat{r}_N, \hat{z}(L-N-1), \ldots, \hat{z}(0)\} \qquad (29)$$

When the signal extension support is not needed, $\mathbf{P}_L$ is replaced with $\mathbf{P}$ and $\mathbf{K}$ is replaced with the identity matrix $\mathbf{I}_N$.

6 Results

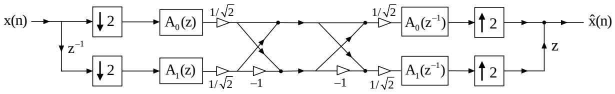


Figure 2: Allpass-based two-channel filter bank.

All results are obtained using a dyadic 6-scale wavelet transform based on the allpass-based two-channel filter bank shown in Figure 2. Allpass filters $A_0(z)$ and $A_1(z)$ are designed to have approximately linear phase (ALP) analysis and synthesis filters [8]. These 2nd order allpass filters have the following transfer functions

$$A_0(z) = \frac{a_0 + a_1 z^1 + a_2 z^2}{a_0 + a_1 z^{-1} + a_2 z^{-2}}, \quad A_1(z) = \frac{a_0 - a_1 z^1 - a_2 z^2}{a_0 - a_1 z^{-1} - a_2 z^{-2}}$$

where $a_0 = 1$, $a_1 = -0.19$ and $a_2 = 0.04$. We are going to compare our results obtained with the conventional circular convolution algorithm [9]. Table 1 shows the peak-signal-to-noise-ratio (PSNR) results of the 512 × 512 Lena 8 bpp greyscale image at various compression ratios encoded with the code-table enhanced version [5] of the SPIHT algorithm [10], for the three methods: circular convolution (CC); embedded filter states without signal extension (EFS), and embedded filter states with (left and right boundary) half-symmetric extension (EFS-HS) support. The results show that on average the EFS, and EFS-HS results are 0.24 dB and 0.41 dB better than the CC results respectively. Compression results using the Daubechies Biorthogonal 9/7 (DB97) wavelet [11] which is selected as the default wavelet transform for lossy image compression by the JPEG 2000 standard is also included for comparison. The results show that EFS-HS implementation produce competitive results against DB97 whose compression results are obtained applying whole-sample symmetric extension.

Impl.	**Compression ratio**				
Method	**8:1**	**16:1**	**32:1**	**64:1**	**128:1**
CC	39.88	36.56	33.32	30.33	27.73
EFS	39.99	36.77	33.61	30.65	27.99
EFS-HS	40.13	36.95	33.82	30.85	28.11
DB97	40.15	37.01	33.90	30.91	28.10

Table 1: Image compression results of the 'Lena' image in terms of PSNR (dB) values for different compression ratios.

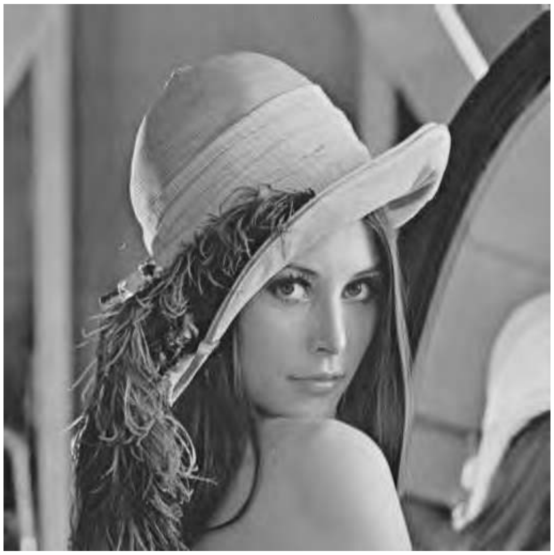
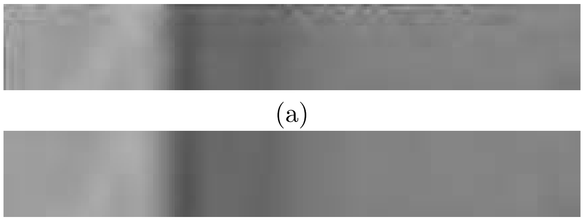


Figure 3: Lena image compressed at 32:1 compression ratio using the EFS-HS algorithm.

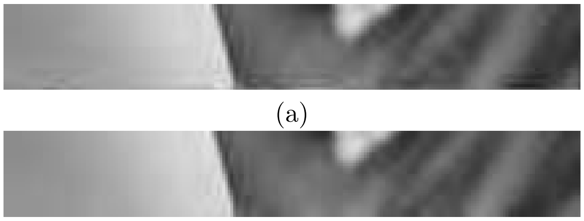
The visual quality of the reconstructed images also plays a major role in the comparison of the filtering algorithms. Figure 4(a), Figure 4(b),Figure 5(a) and Figure 5(b) show the top-left and bottom-right parts extracted from the reconstructed Lena images compressed at 32:1 compression ratio using the CC and EFS-HS algorithms respectively. Visual artefacts due to CC near the top-left and bottom-right borders are clearly seen



(a)

(b)

Figure 4: Top-left parts of the reconstructed Lena images compressed at 32:1 compression ratio using the CC and EFS-HS algorithms.



(a)

(b)

Figure 5: Bottom-right parts of the reconstructed Lena images compressed at 32:1 compression ratio using the CC and EFS-HS algorithms.

in figures 4(a) and 5(a). However if EFS with full signal extension support is employed, then no visual artefacts present in the reconstructed images as shown in figures 4(b) and 5(b).

7 Conclusion

We have presented the EFS method which eliminates the need for transferring the final filter states from the analysis bank to the synthesis bank explicitly. This is achieved by implicitly embedding the final filter states into the analysis output stream in a nonexpansive way. Use of circular convolution also eliminates the need for the transmission of filter states, but causes visual artefacts in the reconstructed images. We have shown that the EFS method with the support for continuity at signal boundaries is superior to circular convolution in both PSNR and visual quality aspects. Another advantage of EFS is that it is as fast and as efficient as the linear convolution in terms of execution speed once the EFS matrices **P** and **K** are initialized. The fraction of difference in performance with the linear convolution is less than %1.

References

[1] John Hakon Husoy and Tor A. Ramstad, "Subband coding of images employing an efficient parallel filter bank," in *Proc. Of SPIE on Visual Communications and Image Processing IV*. Nov. 1989, pp. 752–763, SPIE.

[2] P. P. Vaidyanathan and Tsuhan Chen, "Role of anticausal inverses in multirate filter-banks—part I: System-theoritic fundamentals," *IEEE Trans. on Signal Processing*, vol. 43, no. 5, pp. 1090–1102, May 1995.

[3] Charles D. Creusere and Sanjit K. Mitra, "Image coding using wavelets based on perfect reconstruction IIR filter banks," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. CSVT-6, no. 5, pp. 447–458, Oct. 1996.

[4] Umut Sezen and Stuart S. Lawson, "Anticausal inverses for digital filter banks," in *Proc. ECCTD'01*, Helsinki, Finland, Aug. 2001, vol. I, pp. 229–232.

[5] Umut Sezen, *Anticausal Inverses for IIR Filter Banks, Quincunx Wavelets and Image Coding*, Ph.D. thesis, University of Warwick, April 2003.

[6] Fredrik Gustafsson, "Determining the initial states in forward-backward filtering," *IEEE Trans. on Signal Processing*, vol. 44, no. 4, pp. 988–992, Apr. 1996.

[7] M.E.D. Jimenez and N.G. Prelcic, "Linear boundary extensions for finite length signals and paraunitary two-channel filterbanks," *IEEE Trans. on Signal Processing*, vol. 52, no. 11, pp. 3213–3226, Nov. 2004.

[8] Stuart S. Lawson and A. Klouche-Djedid, "Technique for design of two-channel approximately linear phase QMF filter bank and its application to image compression," *IEE Proc. on Vision Image and Signal Processing*, vol. 148, no. 2, pp. 85–92, Apr. 2001.

[9] Mark J. T. Smith and Steven L. Eddins, "Analysis/Synthesis techniques for subband image coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 8, pp. 1446–1456, Aug. 1990.

[10] Amir Said and William A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.

[11] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using the wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.