

Adaptive Predictor Based on Maximally Flat Halfband Filter in Lifting Scheme

Wen-Jen Ho and Wen-Thong Chang

Abstract—For the complex short time-varying signals, a high-order predictor does not always yield good performance. For this, we investigate the use of a short-order adaptive predictor. Since the maximally flat filters are the optimal predictors for polynomial signal prediction, the adaptation is based on the combination of a set of maximally flat filters. For compression efficiency, the dynamic ranges of the weighting variables are specially considered. For this, based on the Bernstein filters, another form to represent the weighting variables is used. These two sets of weighting coefficients can be transformed into each other with a simple linear transform. Thus, the adaptation can be made in both the time domain and the frequency domain. For block-based image coding, the least square criterion is used to derive the weighting coefficients. Experimental results show that the adaptive predictor performs better than the S+P transform, the median edge detector (MED), and the gradient adjusted predictor (GAP).

Index Terms—Bernstein polynomial, filter bank, lifting scheme, maximally flat filter.

I. INTRODUCTION

APPLICATIONS of filter banks for lossless image coding have been shown in [1]–[6]. In these approaches, the filtered outputs are truncated to result in an integer to integer transform. The S transform is the simplest way of obtaining the integer filtered samples. It can be seen as the integer implementation of the Haar wavelet transform. To improve the coding efficiency, Said and Pearlman extended the S transform to the S+P transform (S transform + prediction) [1], where a prediction stage to predict the highpass component from the lowpass component was added. In fact, the S transform and S+P transform can be seen as the special cases of the lifting scheme [2], [7], [8]. A more general approach to construct filter banks with a lifting scheme that maps integers to integers can be seen in [2]. To reduce the levels of the gray value, downward or upward truncation is used [5].

One lifting step consists of applying a predictor to the even (or odd) samples and subtracting the result from the odd (or even) samples. Eventually, after several lifting steps, equivalent lowpass and highpass components can be obtained. In each lifting step, the predictor output is downward truncated to result in an integer-to-integer transform. In the lifting scheme, the same downward truncation is used in both the analysis and

synthesis processes. On the contrary, in the rounding transform [3], the downward and upward truncations are used in the forward and backward transform, respectively. The rounding transform is very similar to the lifting scheme, where the difference is that the lifting scheme is defined on the time domain, but the rounding transform is defined on the Z domain.

For practical image coding, no filters can perform consistently better than others. Some commonly used design parameters such as the order of vanishing moments, the filter tap length, etc., are not sufficient to specify the filter for practical applications. For efficient coding, minimization of the highpass components is a commonly used method. For complex short-time varying signals, a high-order predictor does not always yield good performance. For this, we investigate the use of the short-order adaptive predictor. In this paper, we focus on the design method of the adaptive predictor to be used in the lifting scheme. The predictor used in the lifting scheme is a halfband filter. Thus, an adaptive halfband filter is used with the criterion to minimize the energy of the highpass components. The primal lifting step [7] that updates the lowpass components are kept fixed. In the extreme case, if the primal lifting is neglected, the delta function becomes the analysis lowpass filter and the halfband filter is the synthesis lowpass filter. The adaptation is done on a block-by-block basis to make the highpass output of such a filter bank as small as possible.

The reason to consider such a structure is that the prediction with the halfband filter with direct decimated signals possesses some well-known properties. The tap length and the zero distribution of the filter are related to the order of the polynomials to be interpolated. This is the so-called A_p condition mentioned in [9]. An immediate consequence of this theorem is that the highpass components of the filter banks will be zero if the signals to be processed are polynomial of degree $N - 1$ and the filter has N zeros at π . In this particular case, the optimal predictor to minimize the highpass components of the filter bank is the maximally flat (Maxflat) filter [10]–[12]. To extend such a theorem to the practical applications, an adaptive filter based on the linear combination of a set of maximally flat filters is considered. Since either the actual signals are often time varying within a short interval such that a single filter cannot be used to predict it well or the order of the signal to be predicted is higher than the order of the filter, the actual signal cannot be reproduced.

Based on the A_p condition, an adaptive halfband filter can be described as a linear combination of the Maxflat filters. The weightings of the combination can be used to describe the image statistics in the time domain. For compression, the dynamic ranges of the weightings are important. To enable

Manuscript received July 17, 1998; revised April 30, 1999. This work was supported by the National Science Council, Taiwan, R.O.C., under Contracts NSC-85-2213-E-009-018 and NSC-88-2213-009-017. The associate editor coordinating the review of this paper and approving it for publication was Dr. Hitoshi Kiya.

The authors are with the Department of Communication Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

Publisher Item Identifier S 1053-587X(99)08300-2.

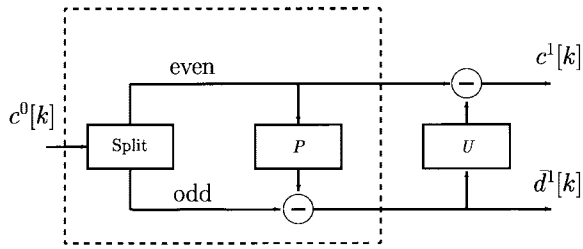


Fig. 1. Lifting scheme.

such a design, the halfband filter is also expressed as a linear combination of the Bernstein filter. The Bernstein filter will be discussed in detail in Section III. The weightings of the linear combination of the Bernstein filter can be used to describe the image statistics in the frequency domain. There exists a linear transform to map the representation space from the Bernstein bases to the Maxflat bases.

To capture the short time change of the signal, block-based adaptation is considered. For each block of signals, one predictor to minimize the highpass components is used. In our experiments, the adaptive lifting scheme performs better than that of the nonadaptive S and S+P transform. Further, our proposed adaptive predictor is comparable with the median edge detector and gradient adjusted predictor, which are used in the LOCO-I and CALIC, respectively [13]–[15]. However, LOCO-I and CALIC are single-resolution predictive coding schemes that do not have progressive decoding capability, as our proposed method does. Besides the prediction, a complete coding scheme consisting of the prediction \bar{d}^j , error modeling, and set partitioning is also presented.

The organization of this paper is as follows. In Section II, the basic structure considered in this paper and its relations with lifting scheme are introduced. According to the approximation property of the wavelet representation, polynomial interpolation with regular halfband filters is briefly described. In Section III, application with a time-varying signal is discussed. Two forms of parameterized halfband filters are briefly described. The adaptive interpolation algorithm based on the parameterized halfband filters is also presented. In Section IV, the experimental results are shown. Finally, in Section V, a conclusions is made.

II. INTERPOLATIVE DPCM PYRAMID

This section provides an introduction to the multirate structure considered in this paper. The lifting scheme is a flexible technique for construction of a wavelet through a series of lifting steps. As shown in Fig. 1, the lifting scheme first splits a signal into its even and odd samples. Then, alternating primal and dual lifting steps are used to process the decimated samples. A dual lifting step consists of applying a filter P to the even samples and subtracting the result from the odd ones. A primal lifting step consists of applying a filter U to the odd samples and subtracting the result from the even samples. Eventually, after several lifting steps, the even samples will become the lowpass components, whereas the odd samples become the highpass components. In each lifting step, the filter output is truncated. This results in an integer-to-integer

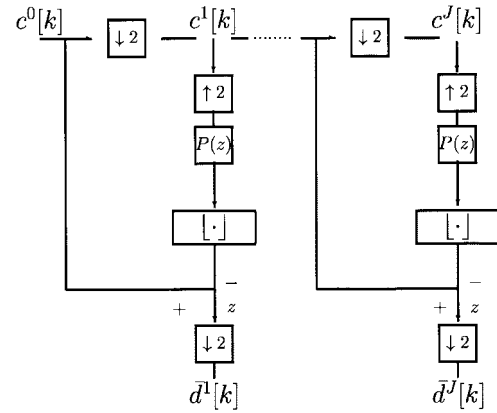


Fig. 2. Decomposition structures of the interpolative DPCM pyramid.

transform. The dual lifting step can be regarded as a prediction stage to predict the highpass components, and the primal lifting step can be regarded as an update stage to update the lowpass components. The basic concepts of the dual and primal lifting steps are the same. In this paper, the design of the predictor to be used in the lifting scheme is considered. We focus on the dual step with the constraint to minimize the highpass energy based on the known lowpass components.

The implementation of the integer dual lifting scheme is illustrated in Fig. 2. Formally, a top-down decomposition procedure can be described as, for $j = 1 \sim J$

$$\begin{aligned} c^j[k] &= c^{j-1}[2k] \\ \bar{d}^j[k] &= d^{j-1}[2k+1] \\ &= c^{j-1}[2k+1] - \left[\sum_n c^j[n] p[2n - (2k+1)] \right] \\ &= c^{j-1}[2k+1] - \left[\sum_n c^j[n] e^1[n-k-1] \right] \end{aligned} \quad (1)$$

where $e^1[k] = p[2k+1]$ is the first polyphase component of $p[k]$. For reconstruction, $c^j[k]$ can be obtained with prediction from its lower resolution versions plus the prediction errors. A bottom-up reconstruction procedure is, for $j = J \sim 1$

$$\begin{aligned} c^{j-1}[2k] &= c^j[k] \\ c^{j-1}[2k+1] &= \bar{d}^j[k] + \left[\sum_n c^j[n] p[2n - (2k+1)] \right] \\ &= \bar{d}^j[k] + \left[\sum_n c^j[n] e^1[n-k-1] \right]. \end{aligned} \quad (2)$$

Note that the sequences $c^j[k]$, $j = 1 \sim J$ can be obtained concurrently when sequence $c^0[k]$ is given; therefore, all the prediction errors $\bar{d}^j[k]$, $j = 1 \sim J$ can be computed in parallel [16]. Since $\bar{d}^j[k]_{j=1 \sim J}$ are the differences between the interpolator's output and the original signal, this scheme is similar to that of the interpolative DPCM (IDPCM) [17], [18]. Since the interpolation process is executed iteratively to obtain the prediction errors with a pyramid structure, for convenience, the whole system based on the dual lifting scheme is called the interpolative DPCM (IDPCM) pyramid.

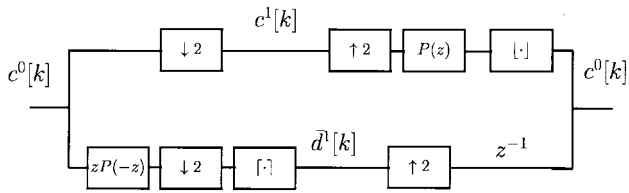


Fig. 3. Degenerate filter bank.

Except for the truncation process, this kind of hierarchical interpolation has also been used in [19]–[22]. In [19] and [20], the simple bilinear interpolation is used. The compression ratios are shown to be higher than DPCM and in the range of DCT under equivalent test conditions [19]. In [22], a separable first-order interpolation is used, and entropy-constrained trellis coded quantization (ECTCQ) is used to code the prediction errors. The performance is claimed to be superior to that of JPEG.

In fact, one critically subsampled pyramid, as shown in Fig. 2, is exactly equivalent to one critically subsampled filter bank [23]. The filter bank is shown in Fig. 3. In analogy with the lossless DPCM, the integer highpass component can be obtained similarly by truncating the noninteger number, as shown in Fig. 3. The symbol $\lceil \cdot \rceil$ is the upward truncation, and the symbol $\lfloor \cdot \rfloor$ is the downward truncation. In the IDPCM pyramid, the same downward truncation is used in both the decomposition and reconstruction processes. On the contrary, in the filter bank, the upward and downward truncations are used in the analysis and synthesis parts, respectively. If the truncation is neglected, an associated set of biorthogonal bases exists. The scaling function is called the interpolative scaling function since $\Phi(t)$ is cardinal, i.e., $\Phi(k) = \delta(k), k \in \mathbb{Z}$. According to the two-scale relation

$$\Phi(t) = \sum_k p[k] \Phi(2t - k) \quad (3)$$

the values of $\Phi(t)$ on the half-integer points can be computed by $\Phi(m/2) = \sum_k p[k] \Phi(m - k)$. Since $\Phi(k) = \delta(k)$, $\Phi(k/2) = p[k]$. This means that the filter $p[k]$ is the sampled sequence of the scaling function sampled at half integer points. There is a close connection between a pair of biorthogonal scaling functions and an interpolative scaling function [24], [25]. It is easy to find that the wavelet is $\Psi(t) = \Phi(2t - 1)$. This means that the wavelet is a translated and dilated version of the scaling function itself. The dual scaling function is the delta function $\delta(t)$, and the dual wavelet is a linear combination of the delta function, i.e., $\tilde{\Psi}(t) = \sum_k (-1)^{k+1} p[k+1] \delta(2t - k)$. Strictly speaking, the delta function $\delta(t)$ is not defined for any t . However, since they can satisfy the framework of the biorthogonal wavelet transform, the functions $\tilde{\Phi}(t)$ and $\tilde{\Psi}(t)$ are still named the dual scaling function and the dual wavelet, respectively. A thorough treatment on the use of interpolating wavelet transform can be found in [16]. This kind of transform is claimed to be optimal from the point of view of computing individual coefficients in parallel since the coefficients are obtained from linear combination of samples rather than integrals.

A. Accuracy of Approximation

The primary concern of this paper is the design of the predictor. In this section, we discuss the relation between the interpolative scaling function and the polynomial function. The purpose is to indicate the necessary condition for polynomial interpolation. More important, we will discuss the case when the polynomial function is a time-varying one.

In the wavelet transform, a function is projected onto a space hierarchy V_j . The index j gives the time scale information $\Delta t = 2^j$. In practice, the level j is determined by balancing between the accuracy and the cost. The cost is approximately doubled from one level to the next since the number of the basis functions and the coefficients is doubled. The representation accuracy depends both on the scaling functions $\Phi(t)$ and the signal $f(t)$. In the following, the accuracy of approximation with the interpolative wavelet representation for a continuous function is analyzed.

Consider a continuous function $f(t)$ with derivative $f^{(i)}(t), 0 \leq i \leq N$. It can be shown that if $\Phi(t)$ satisfies the condition

$$\sum_{k=-\infty}^{\infty} (t-k)^m \Phi(t-k) = \delta(m), \quad m = 0 \sim N-1 \quad (4)$$

then $\|I_j f(t) - f(t)\|_c \leq (M2^j N / N!) \|f^{(N)}(t)\|_c$, where M is a constant, and the norm $\|f(\cdot)\|_c = \sup_{t \in R} |f(t)|$ [26]. The term $(M2^j N / N!) \|f^{(N)}(t)\|_c$ is the upper bound of the approximation error when the basis $\Phi(t)$ satisfying (4) is used. To derive this result, let us apply the Taylor series expansion to the function $f(u)$ with respect to the point $t \in R$. That is

$$f(u) = \sum_{m=0}^{N-1} \frac{f^{(m)}(t)}{m!} (u-t)^m + \frac{f^{(N)}(\eta)(u-t)^N}{N!} \quad (5)$$

where η is within the interval between u and t .

When the function $f(t)$ is sampled with a set of discrete points at $u = 2^j k$, $I_j f(t)$ is defined as

$$\begin{aligned} I_j f(t) &= \sum_{k=-\infty}^{\infty} f(2^j k) \Phi(2^{-j} t - k) \\ &= \sum_{k=-\infty}^{\infty} \left[\sum_{m=0}^{N-1} \frac{f^{(m)}(t)}{m!} (2^j k - t)^m + \frac{f^{(N)}(\eta)(2^j k - t)^N}{N!} \right] \Phi(2^{-j} t - k) \\ &= \sum_{m=0}^{N-1} \frac{f^{(m)}(t)}{m!} \sum_{k=-\infty}^{\infty} (2^j k - t)^m \Phi(2^{-j} t - k) \\ &\quad + \frac{1}{N!} \sum_{k=-\infty}^{\infty} f^{(N)}(\eta) (2^j k - t)^N \Phi(2^{-j} t - k) \\ &= f(t) \sum_{k=-\infty}^{\infty} \Phi(2^{-j} t - k) \\ &\quad + \sum_{m=1}^{N-1} \frac{f^{(m)}(t)}{m!} \sum_{k=-\infty}^{\infty} (2^j k - t)^m \Phi(2^{-j} t - k) \\ &\quad + \frac{1}{N!} \sum_{k=-\infty}^{\infty} f^{(N)}(\eta) (2^j k - t)^N \Phi(2^{-j} t - k). \quad (6) \end{aligned}$$

Since $\sum_{k=-\infty}^{\infty} (2^j k - t)^m \Phi(2^{-j} t - k)$ can be rewritten as $(-1)^m 2^{jm} \sum_{k=-\infty}^{\infty} (t - k)^m \Phi(t - k)$, by (4), we have

$$I_j f(t) = f(t) + \frac{(-1)^N 2^{jN}}{N!} \sum_{k=-\infty}^{\infty} f^{(N)}(\eta)(t - k)^N \Phi(t - k) \quad (7)$$

and

$$\begin{aligned} \|I_j f(t) - f(t)\|_c &\leq \frac{M 2^{jN}}{N!} \|f^{(N)}(t)\|_c \\ &\approx C(\Delta t)^N \|f^{(N)}(t)\|_c \end{aligned} \quad (8)$$

holds, where $M = \sup_{t \in \mathbb{R}} \sum_k |t - k|^N |\Phi(t - k)|$, and $C = (M/N!)$. The constant C and the exponent N depend on the choice of $\Phi(t)$. When constructing $I_j f(t)$, the step from $\Delta t = 2^j$ to $\Delta t = 2^{j-1}$ divides the approximation error by about 2^N . Thus, the number N is critical. Usually, the constant C is less critical. The norm $\|f^{(N)}(t)\|_c$ of the N th derivative depends on the signal $f(t)$. If the signal $f(t)$ is smooth, then $f^{(N)}(t)$ is small, and the error will be small. If the signal $f(t)$ contains abrupt changes, then $f^{(N)}(t)$ is large, and the error will be large.

What (8) indicates is that if the function $f(t)$ is a polynomial of degree less than or equal to $N - 1$, the interpolated signal $I_j f(t)$ will reproduce it if $\Phi(t)$ satisfies (4). This fact has been indicated in [9]. However, if the function $f(t)$ is a polynomial with degree larger than $N - 1$, the approximation error will be dominated by $(M 2^{jN}/N!) \|f^{(N)}(t)\|_c$. We will discuss this case later.

Now, let us concentrate on some basic properties of the basis. With the two-scale relation shown in (3), the property of the basis shown in (4) can be related to the property of the filter $p[k]$. Thus, the accuracy of the interpolation can be determined from the property of the filter. The highest degree of the interpolated polynomial can be determined from $p[k]$ or from $P(e^{jw})$. With the Poisson summation formula [26]

$$\sum_k (t - k)^m \Phi(t - k) = \sqrt{2\pi} \sum_k (-j)^m \hat{\Phi}^{(m)}(2\pi k) e^{j2\pi k t} \quad (9)$$

can be obtained, where $\hat{\Phi}(w)$ denotes the Fourier transform of $\Phi(t)$. Therefore, the condition in (4) is also equivalent to

$$\hat{\Phi}^{(m)}(2\pi k) = \delta(k) \delta(m), \quad m = 0 \sim N - 1. \quad (10)$$

This means that $\hat{\Phi}(w)$ must have zeros of order N at all frequencies $w = 2\pi k, k \neq 0$. This is the so-called Strang–Fix condition [9]. Therefore, (4) can be regarded as the Strang–Fix condition in the time domain.

Thus, if $\hat{\Phi}(w)$ satisfies the Strang–Fix condition, then the corresponding requirement on the filter $p[k]$ is that it has N zeros at π , i.e., the transfer function $P(e^{jw})$ can be represented as

$$P(e^{jw}) = (1 + e^{jw})^N R(e^{jw}) \quad (11)$$

where $R(e^{j\pi}) \neq 0$. A filter with N zeros at $w = \pi$ is a N -regular filter. With the N -regular filter, the corresponding scaling function and its translates reproduce polynomials up to degree $N - 1$. From the time-domain Strang–Fix condition, this fact can also be seen. Condition (4) can be rewritten in

the discrete form as

$$\sum_k \left(\frac{m}{2} - k\right)^n p[m - 2k] = \delta[n], \quad n = 0 \sim N - 1 \quad (12)$$

if $t = (m/2)$ is considered. This property is the sum rule $\sum_k (-1)^k k^n p[k] = 0, n = 0, \sim N - 1$ of the halfband filter. This is the A_p condition mentioned in [9], indicating that there are N zeros at $w = \pi$.

The above-mentioned Strang–Fix condition indicates that an N -regular halfband filter can be used to interpolate a polynomial of degree $N - 1$. The minimum length of an N -regular halfband filter is $2N - 1$ and is called the maximally flat filter. The above derivation shows the results that can be obtained when the maximally flat filter is used for interpolation. What (6) indicates is how the sampled points from $f(t)$ can be used to interpolate the same polynomial function $f(t)$.

For practical application, the signal is usually a time- or space-varying one. The order of the underlying polynomial is difficult to predict in advance. For abrupt changing signals, we are often faced by the case of insufficient sampled points. In this situation, perfect polynomial reconstruction is not possible because the number of zeros of the filter at π is less than the order of the polynomial. This has been mentioned briefly above when the degree of the polynomial is larger than $N - 1$ and the filter has only N zeros at π . Thus, for practical application with a filter of fixed length, the prediction error is usually due to the above-mentioned phenomena. Since the error cannot be avoided with a single filter bank, we then seek to use an adaptive filter bank to minimize the prediction error. The easiest way is to use a degenerate filter bank with variable halfband filters.

III. PARAMETERIZED HALF BAND FILTERS

As mentioned in Section II-A, for a polynomial up to degree $N - 1$, the N regular halfband filter $p[k]$ can be used to recover it from its sampled values. Let us denote this class of polynomials as Θ_{N-1} . Thus, with an interpolative filter bank, if $c^j[k] = f(k/2^j)$, where $f(t) \in \Theta_{N-1}$, the highpass components $\bar{d}^{j+1}[k] = 0$ for any j . The function $f(t)$ can be recovered by iterative interpolation from its samples $c^j[k]$ using the same filter $p[k]$ at each stage. Since the filter $P(z)$ has at least N zeros at π , the minimum length of such a filter is $2N - 1$. Thus, the minimum required number of sampled points needed to reconstruct the polynomial is N .

A filter with most its zeros at π is a maximally flat (Maxflat) filter. The Maxflat filter is the optimal minimum length predictor for polynomial signals. To see the effect of Maxflat filters on the results of polynomial interpolation, let us first show an example: $f^1(t) = (t-1)^3 - (t-1)$. The sampled sequences of the cubic polynomial $f^1(t)$ with $\Delta t = \frac{1}{64}$ and $\Delta t = \frac{1}{4}$ are shown in Fig. 4(a) and (b). The filter used for interpolation is $P(z) = (z^2/16)(1 + z^{-1})^4(-z + 4 - z^{-1})$, which is a seven-tap Maxflat filter. As discussed in Section II-A, any filter with more than four zeros at π can be used to interpolate $f^1(t)$. The corresponding interpolated sequence after four iterations with this filter is shown in Fig. 4(c). The errors between the interpolated signal and the signal $f^1(k/64)$

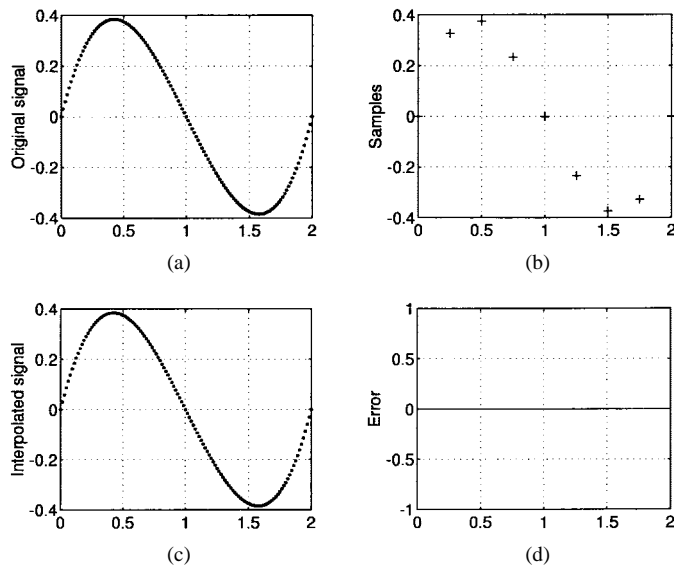


Fig. 4. Interpolation of the polynomial $f^1(t)$. (a) Original sampled sequence with $\Delta t = \frac{1}{64}$. (b) Original sampled sequence with $\Delta t = \frac{1}{4}$. (c) Interpolated signal with the seven-tap Maxflat filter. (d) Errors between the original signal and the interpolated signal.

are plotted in Fig. 4(d). As can be seen, the interpolated result with the Maxflat filter is the same as $f^1(k/64)$.

Now, let us consider two practical situations. The first is when the degree of the polynomial is larger than $N - 1$. That is, the sampled points are less than N . The second case corresponds to a time-varying signal. For these two cases, perfect polynomial reproduction cannot be obtained. Consider the two test signals

$$f^2(t) = (t - 1)^5 - (t - 1) \quad \text{and}$$

$$f^3(t) = \begin{cases} t^3, & t \leq 0.5 \\ 2t^3 - 0.13, & 0.5 < t \leq 1 \\ t^3 + t - 0.07, & 1 < t \leq 1.5 \\ 0.33t^3 - 0.87t + 5.64, & 1.5 < t. \end{cases}$$

The results interpolated with the seven-tap Maxflat filter are shown in Figs. 5 and 6. As shown in these figures, there exist errors in the interpolated results. The function $f^3(t)$ is a time-varying signal.

To overcome such problems, in this paper, a combined interpolation scheme is proposed. That is, interpolation with multiple Maxflat filters is proposed. The result of the interpolation is the linear combination of the results from all the Maxflat filters. The idea is to treat the above-mentioned two cases as signals composed of polynomials of many different orders. Thus, in the prediction process, along the time or space domain, each point to be interpolated will be characterized by a set of parameters corresponding to the weightings for all the Maxflat filters used. From the coding point of view, the coding efficiency will depend on the distributions of these weighting parameters. In order to maximize the coding gain, block-based prediction is considered. That is, we seek a common set of parameters for a block of signals such that the total prediction errors are minimized.

A linear combination of the Maxflat filters with different orders will no longer be a Maxflat filter. Thus, for the block-

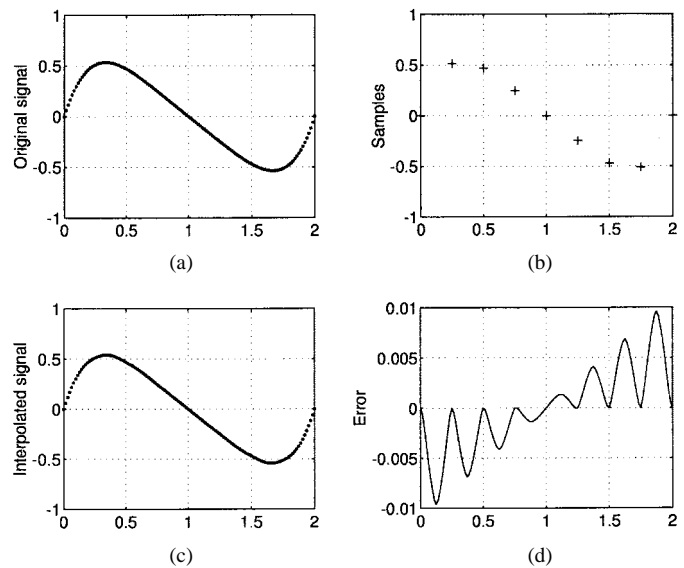


Fig. 5. Interpolation of the polynomial $f^2(t)$. (a) Original sampled sequence with $\Delta t = \frac{1}{64}$. (b) Original sampled sequence with $\Delta t = \frac{1}{4}$. (c) Interpolated signal with the seven-tap Maxflat filter. (d) Errors between the original signal and the interpolated signal.

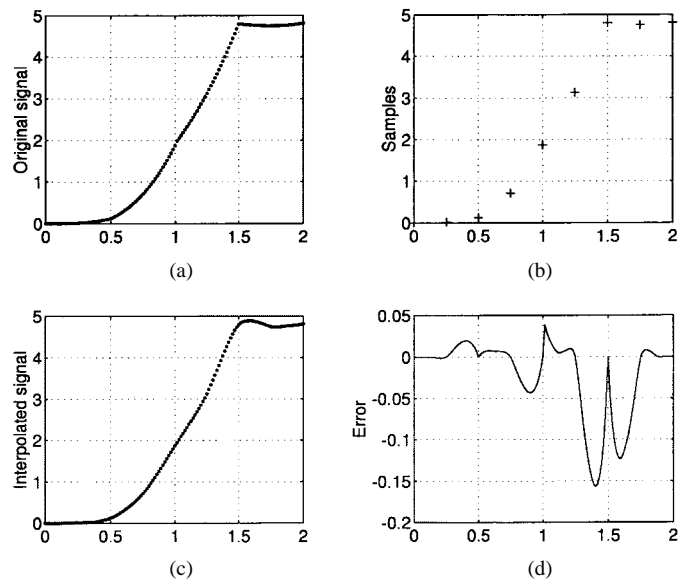


Fig. 6. Interpolation of the polynomial $f^3(t)$. (a) Original sampled sequence with $\Delta t = \frac{1}{64}$. (b) Original sampled sequence with $\Delta t = \frac{1}{4}$. (c) Interpolated signal with the seven-tap Maxflat filter. (d) Errors between the original signal and the interpolated signal.

based predictive coding, we are considering a non-Maxflat halfband filter. There are two forms that can be used to represent the non-Maxflat halfband filter. The first form is the direct linear combination of Maxflat filters of different lengths. This form is simple, but the distributions of the weighting parameters from many different blocks are quite large. For this, other bases derived from Bernstein polynomials are used [27], [28]. These basis filters can be called Bernstein filters. A non-Maxflat filter can be described as the linear combination of the Bernstein filters. In the following, we first discuss the property of the Bernstein filter. Then, the transformation between the two forms will be discussed.

A. Filter Design Based on Bernstein Polynomials

The n th-order Bernstein polynomials are defined as

$$b_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i}, \quad i = 0, \dots, n, \quad (13)$$

$$0 \leq t \leq 1$$

where the binomial coefficients $\binom{n}{i}$ are given by

$$\binom{n}{i} = \begin{cases} \frac{n!}{i!(n-i)!}, & 0 \leq i \leq n \\ 0, & \text{otherwise.} \end{cases}$$

The Bernstein polynomials have the following properties: $b_0^n(0) = 1$, $b_n^n(1) = 1$, and $\sum_{i=0}^n b_i^n(t) = 1$ [29].

Given $n+1$ equally spaced sampled values μ_i , $0 \leq i \leq n$, over the interval $[0, 1]$, the n th-order approximation $B_n(t)$ based on Bernstein polynomials is defined as

$$B_n(t) = \sum_{i=0}^n \mu_i b_i^n(t), \quad 0 \leq t \leq 1. \quad (14)$$

The samples μ_i are the weighting coefficients of the approximation $B_n(t)$. With the transform $t = \sin^2(w/2)$ [11], [27], the function $B_n(t)$ over the interval $[0, 1]$ can be transformed into the frequency response $\frac{1}{2}P(e^{jw})$ of a filter over the interval $[0, \pi]$.

It has been shown in [27] that to design a halfband filter, the weighting coefficients μ_i can be set as

$$\mu_i = \begin{cases} 1 - \alpha_i, & 0 \leq i \leq N-1 \\ \alpha_i, & N \leq i \leq 2N-1 \end{cases}$$

with $\alpha_i = \alpha_{2N-1-i}$. With such design parameters, the halfband condition can be achieved. That is

$$B_{2N-1}(t) + B_{2N-1}(1-t) = 1, \quad 0 \leq t \leq 1. \quad (15)$$

The approximation function $B_{2N-1}(t)$ can be transformed to the function $\frac{1}{2}P(e^{jw})$, and the approximation function $B_{2N-1}(1-t)$ can be transformed to the function $\frac{1}{2}P(e^{j(\pi-w)})$. Therefore, according to (15), the filter $P(e^{jw})$ transformed from $B_{2N-1}(t)$ is a halfband filter.

Using the mapping $\cos w = (z + z^{-1})/2$, $P(e^{jw})$ can be transformed into the transfer function $P(z)$ with $4N-1$ coefficients in the z domain. However, among the $4N-1$ coefficients, only N of them need be specified due to the symmetric halfband condition. With this mapping, $P(z)$ is then obtained as

$$P(z) = z^{2N-1} \frac{(1+z^{-1})^{2N}}{2^{4N-3}} \left\{ \sum_{i=0}^{N-1} (-1)^i \binom{2N-1}{i} \cdot (1+z^{-1})^{2(N-1-i)} (1-z^{-1})^{2i} \right. \\ \left. - \sum_{i=0}^{N-1} (-1)^i \alpha_i \binom{2N-1}{i} \right. \\ \left. \cdot (1+z^{-1})^{2(N-1-i)} (1-z^{-1})^{2i} \right. \\ \left. + \sum_{i=N}^{2N-1} (-1)^i \alpha_{2N-1-i} \binom{2N-1}{i} \right. \\ \left. \cdot (1+z^{-1})^{2(N-1-i)} (1-z^{-1})^{2i} \right\}. \quad (16)$$

This is the halfband filter with length $4N-1$ with α_i as the design parameter. To derive the Bernstein filter, the equation

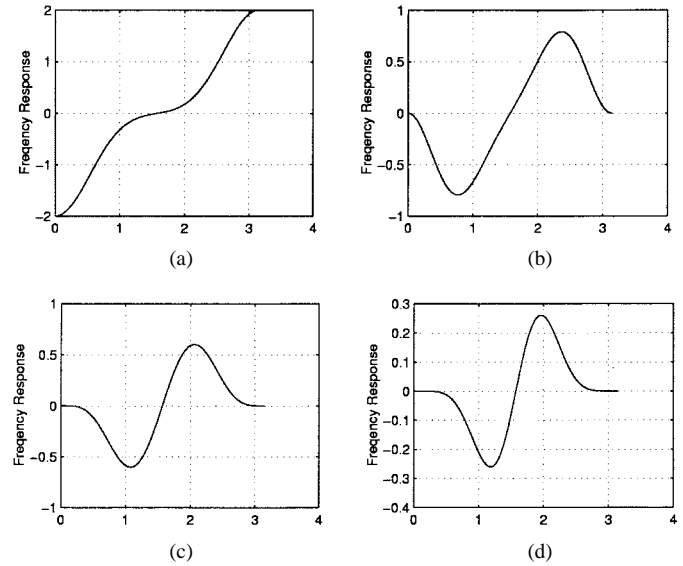


Fig. 7. Frequency responses of the Bernstein filters (a) $B_0^4(z)$. (b) $B_1^4(z)$. (c) $B_2^4(z)$. (d) $B_3^4(z)$.

can be rewritten as

$$P(z) = \sum_{k=0}^{4N-2} \left\{ \sum_{i=0}^{N-1} \binom{2N-1}{i} \sum_{j=0}^k \frac{(-1)^{i+j}}{2^{4N-3}} \cdot \binom{4N-2-2i}{2k-j} \binom{2i}{j} \right\} z^{2N-1-k} \\ + \sum_{i=0}^{N-1} \alpha_i \left(\sum_{k=0}^{4N-2} \left\{ \frac{1}{2^{4N-3}} \binom{2N-1}{i} \cdot \left[\sum_{j=0}^k (-1)^{2N-1-i+j} \binom{2i}{2k-j} \binom{4N-2-2i}{j} \right. \right. \right. \\ \left. \left. \left. - \sum_{j=0}^k (-1)^{i+j} \binom{4N-2-2i}{2k-j} \binom{2i}{j} \right] \right\} z^{2N-1-k} \right) \\ = P_N(z) + \sum_{i=0}^{N-1} \alpha_i B_i^N(z) \quad (17)$$

where the filter $P_N(z)$ is a Maxflat filter with $2N$ zeros at $z = -1$, and the filter $B_i^N(z)$ is the associated i th Bernstein filter with length $4N-1$. These Bernstein filters are also halfband filters. As an example, the frequency responses of the 15-tap Bernstein filters $B_i^4(z)$, $i = 0 \sim 3$ are shown in Fig. 7. There will be N Bernstein filters associated with a maxflat halfband filter with length $4N-1$. The Bernstein filters are bandpass filters with peak frequencies at $\cos^{-1}(2N-2i-2/2N-1)$, $i = 0 \sim N-1$. It can be seen from these figures that the frequency response of the Bernstein filter such as $B_i^N(e^{jw})$, $i = 0 \sim N-1$ has only one peak over the interval $[0, (\pi/2)]$. The advantage to such a representation as shown in (17) is that only one of the parameters α_i needs be adjusted if a portion of the frequency response $P(e^{jw})$ is to be specified. Therefore, the parameters α_i control how each part of the frequency regions is to be adapted to the signals. Thus, the linear combination of the Bernstein filters can be viewed as a frequency-domain design method.

With the halfband condition in (16), we have $p[2k] = \delta[k]$. Let us denote $e^1[k] = p[2k+1]$. With the first polyphase component $e^1[k]$ in the vector form, i.e., $\mathbf{e}^1 = (e^1[-N], e^1[-N+1], \dots, e^1[N-1])^T$, (16) can be rearranged as

$$\mathbf{e}^1 = \mathbf{e}_N + \mathbf{F}\bar{\alpha} \quad (18)$$

where

$$\mathbf{F} = \begin{bmatrix} w_{0,0} & w_{0,1} & \cdots & w_{0,N-1} \\ w_{1,0} & w_{1,1} & \cdots & w_{1,N-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{2N-1,0} & w_{2N-1,1} & \cdots & w_{2N-1,N-1} \end{bmatrix}$$

$$w_{k,i} = \frac{1}{2^{4N-3}} \binom{2N-1}{i} \cdot \left[\sum_{j=0}^{2k} (-1)^{2N-1-i+j} \binom{2i}{2k-j} \binom{4N-2-2i}{j} - \sum_{j=0}^{2k} (-1)^{i+j} \binom{4N-2-2i}{2k-j} \binom{2i}{j} \right]$$

$$k = 0 \sim 2N-1, \quad i = 0 \sim N-1. \quad (19)$$

The vector $\mathbf{e}_N = (e_N[-N], e_N[-N+1], \dots, e_N[N-1])^T$ corresponds to the first polyphase component of the Maxflat filter. It has $2N$ zeros at π . Its element $e_N[k]$ is given in the form

$$e_N[-N+k] = \sum_{i=0}^{N-1} \binom{2N-1}{i} \sum_{j=0}^{2k} \frac{(-1)^{i+j}}{2^{4N-3}} \cdot \binom{4N-2-2i}{2k-j} \binom{2i}{j}$$

$$k = 0 \sim 2N-1. \quad (20)$$

The vector $\bar{\alpha} = (\alpha_0, \alpha_1, \dots, \alpha_{N-1})^T$ consists of all the design parameters $\alpha_i, i = 0 \sim N-1$. If all the $\alpha_i, i = 0 \sim N-1$ are set to zero, the filter becomes a Maxflat filter.

B. Non-Maxflat Filter

In the beginning of this section, we have mentioned the design of a non-Maxflat filter by linear combination of a set of Maxflat filters. Let us first define \tilde{I} as the identity matrix with 1's on the cross diagonal. A 3×3 example is

$$\tilde{I} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

The polyphase matrix that consists of the first polyphase components of a set of Maxflat filters with different lengths is defined as

$$\mathbf{E}_m = \begin{bmatrix} e_N[-N] & 0 & 0 & 0 \\ e_N[-N+1] & e_{N-1}[-N+1] & 0 & 0 \\ \vdots & \vdots & \cdots & e_1[-1] \\ \vdots & \vdots & \cdots & e_1[0] \\ e_N[N-2] & e_{N-1}[N-2] & 0 & 0 \\ e_N[N-1] & 0 & 0 & 0 \end{bmatrix}_{2N \times N}$$

$$= \begin{bmatrix} \mathbf{E}_m^{\text{half}} \\ \tilde{\mathbf{E}}_m^{\text{half}} \end{bmatrix} \quad (21)$$

where $\mathbf{E}_m^{\text{half}}$ denotes the $N \times N$ upper-half matrix of \mathbf{E}_m , and $\tilde{\mathbf{E}}_m^{\text{half}}$ denotes the $N \times N$ lower-half matrix of \mathbf{E}_m . Therefore, the design of a non-Maxflat filter can be described as

$$\mathbf{e}^1 = \mathbf{e}_N + \mathbf{E}_m \bar{\beta} \quad (22)$$

where $\bar{\beta} = [\beta_N, \beta_{N-1}, \dots, \beta_1]^T$ is the weighting vector of the combination.

Now, we discuss the relation between the two forms of the halfband filters and derive the transformation between them. Since the first polyphase components of the Maxflat filters are symmetric, $\tilde{\mathbf{E}}_m^{\text{half}} = \tilde{I} \times \mathbf{E}_m^{\text{half}}$. The matrix $\mathbf{E}_m^{\text{half}}$ is a lower triangular matrix, its inverse matrix always exists, and the following relations hold:

$$I = \mathbf{E}_m^{\text{half}} \times (\mathbf{E}_m^{\text{half}})^{-1}, \quad \tilde{I} = \tilde{\mathbf{E}}_m^{\text{half}} \times (\mathbf{E}_m^{\text{half}})^{-1}. \quad (23)$$

The matrix $(\mathbf{E}_m^{\text{half}})^{-1}$ can be regarded as the transform matrix that transforms the identity matrix I to the polyphase matrix $\mathbf{E}_m^{\text{half}}$. Consider the polyphase component $\mathbf{F}\bar{\alpha}$ in (18). Since the first polyphase components are symmetric, $\mathbf{F}\bar{\alpha}$ can be described as

$$\mathbf{F}\bar{\alpha} = \begin{bmatrix} \mathbf{F}^{\text{half}} \\ \tilde{\mathbf{F}}^{\text{half}} \end{bmatrix} \bar{\alpha}, \quad \text{where } \tilde{\mathbf{F}}^{\text{half}} = \tilde{I} \times \mathbf{F}^{\text{half}}$$

$$= \begin{bmatrix} I \\ \tilde{I} \end{bmatrix} \mathbf{F}^{\text{half}} \bar{\alpha}. \quad (24)$$

The matrix \mathbf{F}^{half} denotes the $N \times N$ upper-half matrix of \mathbf{F} . Substituting (23) into (24), the component $\mathbf{F}\bar{\alpha}$ can be rewritten as

$$\mathbf{F}\bar{\alpha} = \mathbf{E}_m (\mathbf{E}_m^{\text{half}})^{-1} \mathbf{F}^{\text{half}} \bar{\alpha}$$

$$= \mathbf{E}_m \bar{\beta}. \quad (25)$$

Thus

$$\bar{\beta} = [\beta_N, \beta_{N-1}, \dots, \beta_1]^T = (\mathbf{E}_m^{\text{half}})^{-1} \mathbf{F}^{\text{half}} \bar{\alpha}. \quad (26)$$

Equation (26) describes the transform between the two representation forms. Thus, for any non-Maxflat filter, the even-length first polyphase component can be written as the linear combination of the even-length first polyphase components of either a set of Maxflat filters or as a set of Bernstein filters associated with a corresponding Maxflat filter.

For example, consider the case of $N = 2$. From (18), the Bernstein representation form is

$$\mathbf{e}^1 = \frac{1}{16} \left(\begin{bmatrix} -1 \\ 9 \\ 9 \\ -1 \end{bmatrix} + \begin{bmatrix} -1 & 3 \\ -15 & -3 \\ -15 & -3 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \right) \quad (27)$$

where

$$\mathbf{F}^{\text{half}} = \frac{1}{16} \begin{bmatrix} -1 & 3 \\ -15 & -3 \end{bmatrix}.$$

The half-polyphase matrix of the Maxflat filters is

$$\mathbf{E}_m^{\text{half}} = \frac{1}{16} \begin{bmatrix} -1 & 0 \\ 9 & 8 \end{bmatrix}.$$

Its inverse is

$$(\mathbf{E}_m^{\text{half}})^{-1} = \begin{bmatrix} -16 & 0 \\ 18 & 2 \end{bmatrix}.$$

Therefore, the representation with $\bar{\alpha}$ can be transformed to $\bar{\beta}$ as

$$\bar{\beta} = \frac{1}{16} \left(\begin{bmatrix} -16 & 0 \\ 18 & 2 \end{bmatrix} \begin{bmatrix} -\frac{1}{16} & \frac{3}{16} \\ -\frac{13}{16} & \frac{3}{16} \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \right). \quad (28)$$

Thus, we have the following relation:

$$\mathbf{e}^1 = \frac{1}{16} \left(\begin{bmatrix} -1 \\ 9 \\ 9 \\ -1 \end{bmatrix} + (\alpha_0 - 3\alpha_1) \begin{bmatrix} -1 \\ 9 \\ 9 \\ -1 \end{bmatrix} + (-3\alpha_0 + 3\alpha_1) \begin{bmatrix} 0 \\ 8 \\ 8 \\ 0 \end{bmatrix} \right) \quad (29)$$

where the vectors $\frac{1}{16}[-1 \ 9 \ 9 \ -1]^T$ and $\frac{1}{16}[0 \ 8 \ 8 \ 0]^T$ are the first polyphase components of the Maxflat filters with length $N = 2$ and $N = 1$, respectively.

Both forms can be used to describe a general halfband filter. From the coding point of view, the Bernstein form $\bar{\alpha}$ is better due to the smaller dynamic range of α , but the physical meaning of the interpolation in the time domain can be more easily seen with the $\bar{\beta}$ form. In (29), the first term can be seen as the Lagrange interpolation of a cubic polynomial with the Maxflat filter. The second and third terms are the compensation terms. The second term of (29) is also the Lagrange interpolation of a cubic polynomial, but is weighted by $\beta_2 = \alpha_0 - 3\alpha_1$. The third term of (29) is the Lagrange interpolation of a first-order polynomial, i.e., linear interpolation. This term is weighted by $\beta_1 = -3\alpha_0 + 3\alpha_1$. Therefore, interpolation with the non-Maxflat filter is a combination of various odd degree Lagrange filters weighted by β_i .

Therefore, if the point to be interpolated and its surrounding neighbors are on a polynomial Θ_{2N-1} , the Maxflat filter can be used to interpolate it. If the point under consideration is not on the polynomial passing through its surrounding $2N$ known data points, it cannot be recovered by the Maxflat filter. For this, a non-Maxflat filter with appropriate weightings β_i can be used. From the above discussion, it can be seen that the non-Maxflat filter is very useful for general signal interpolation. In the following, we will discuss a block-based coding scheme. Within a block, a common set of parameters is chosen for all the points. This is equivalent to using a filter bank for a block of signals and adjusting the parameters of the filter bank for different blocks. Thus, an adaptation scheme will be discussed in the following section on a block-by-block basis.

C. Adaptive Interpolation

In this section, we discuss the adaptive algorithm used to determine the design parameters of the halfband filter. With the above pyramid decomposition, a block-based coding scheme is considered. The criterion is to minimize the sum of all the prediction errors over all resolution levels. Since α_i and β_i

are linearly related, minimization with respect to α_i is equal to minimization with respect to β_i . In the following, we will discuss the minimization with respect to α_i .

Let us divide the input sequence $c^0[k]$ into blocks with 2^{J+1} samples. In each block, one optimum filter is determined. For mathematical convenience, the error sequences in all the resolution levels $\bar{d}^j[k], j = 1 \sim J$ are described in vector form as

$$\bar{\mathbf{d}} = (\bar{d}^J, \bar{d}^{J-1}, \dots, \bar{d}^1)^T \quad (30)$$

where

$$\bar{d}^j = (\bar{d}^j[0], \bar{d}^j[1], \dots, \bar{d}^j[2^{J+1-j} - 1]). \quad (31)$$

The points $c^{j-1}[2k+1]$ to be predicted are described as

$$\mathbf{y} = (c^{j-1}, c^{j-2}, \dots, c^0)^T \quad (32)$$

where

$$\mathbf{c}^j = (c^j[1], c^j[3], \dots, c^j[2^{J+2-j} - 1]). \quad (33)$$

The corresponding interpolated sequence $\hat{c}^{j-1}[2k+1]$ of $c^{j-1}[2k+1]$ is described in matrix-vector form as

$$\hat{\mathbf{c}} = \mathbf{U}\mathbf{e}^1 \quad (34)$$

where \mathbf{e}^1 , as shown in (18), is the first polyphase component of the halfband filter $p[k]$, and

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_{J,0} \\ \mathbf{u}_{J,1} \\ \vdots \\ \mathbf{u}_{1,2^{J+1-j}-1} \end{bmatrix}$$

where the row vector $\mathbf{u}_{j,k}$ is defined as

$$\mathbf{u}_{j,k} = (c^j[-N-k-1], c^j[-N-k], \dots, c^j[N-k-2]) \\ j = 1 \sim J, \quad k = 0 \sim 2^{J+1-j} - 1. \quad (35)$$

The prediction errors over all resolution levels can be rewritten in matrix-vector form as

$$\begin{aligned} \bar{\mathbf{d}} &= \mathbf{y} - \mathbf{U}\mathbf{e}^1, \\ &= \mathbf{y} - \mathbf{U}(\mathbf{e}_N + \mathbf{F}\bar{\alpha}) \\ &= (\mathbf{y} - \mathbf{U}\mathbf{e}_N) - \mathbf{U}\mathbf{F}\bar{\alpha} \\ &= \tilde{\mathbf{y}} - \tilde{\mathbf{U}}\bar{\alpha} \end{aligned}$$

where $\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{U}\mathbf{e}_N$, $\tilde{\mathbf{U}} = \mathbf{U}\mathbf{F}$. Let $\Theta = \tilde{\mathbf{U}}^T \tilde{\mathbf{U}}$ and $\nu = \tilde{\mathbf{U}}^T \tilde{\mathbf{y}}$. The average squared prediction errors can be formulated as

$$\|\bar{\mathbf{d}}\|_2^2 = \tilde{\mathbf{y}}^T \tilde{\mathbf{y}} - 2\nu^T \bar{\alpha} + \bar{\alpha}^T \Theta \bar{\alpha}. \quad (36)$$

Our objective is to determine the parameters $\bar{\alpha}$ that minimize the average squared error $\|\bar{\mathbf{d}}\|_2^2$.

The gradient vector is the column vector by differentiating (36) with respect to the vector $\bar{\alpha}$

$$\nabla = \frac{\partial(\|\bar{\mathbf{d}}\|_2^2)}{\partial \bar{\alpha}} = 2\Theta \bar{\alpha} - 2\nu. \quad (37)$$

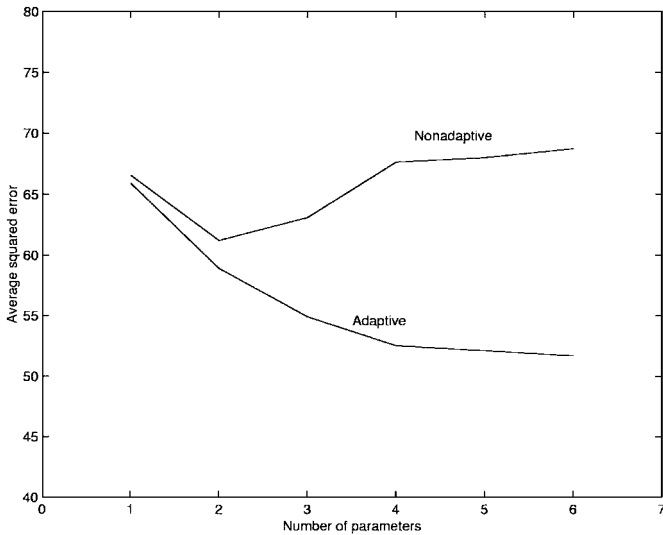


Fig. 8. Average squared errors versus the number of parameters α_i .

Since the squared error is quadratic, the global minimum is obtained when $\nabla = \mathbf{0}$. Setting $\nabla = \mathbf{0}$ in (37) gives the optimum parameters vector $\bar{\alpha}^*$ as

$$\bar{\alpha}^* = \Theta^{-1}\nu. \quad (38)$$

If Θ is positive definite, then $\|\bar{\mathbf{d}}\|_2^2$ attains its minimum value, as in (38). If Θ is not positive definite, an optimal solution can still be determined by replacing the inverse with a pseudo inverse. We can use the least mean square (LMS) algorithm to find the optimal parameters $\bar{\alpha}^*$ with a few iterations [30]. Since the filter is determined based on the known data sequence with LMS algorithm, this optimization is a data-dependent case. The corresponding minimum error is given by

$$\|\bar{\mathbf{d}}\|_2^2 = \tilde{\mathbf{y}}^T \tilde{\mathbf{y}} - \nu^T \bar{\alpha}^*. \quad (39)$$

IV. EXPERIMENTAL RESULTS

In the above sections, we have described the objective of our algorithm. These include the modeling of the signals as polynomials, the optimum filter for polynomial prediction, and the adaptive mechanism for block-based coding of practical signals. In this section, some simulation results are presented. For each block, a single interpolator is used for prediction. The resultant representation for each block consists of the prediction errors hierarchically from 2×2 to 4×4 , from 4×4 to 8×8 , \dots , etc. The values of $\bar{\alpha}^*$ are found by using the adaptive interpolation algorithm described in Section III-C. For each block, extra bits are needed to encode the value of $\bar{\alpha}^*$.

First, we show the need for adaptive prediction. The average squared errors versus the order of the filter for the test image ‘‘Lena’’ are shown in Fig. 8. It can be seen that the average squared error is a monotonically decreasing function of the order of the filter. However, the average squared error saturates when the order of the filter is larger than four. For comparison, the average squared error with the nonadaptive Maxflat filter is also shown. In this case, the average squared error did not decrease with increasing order of the filter. This

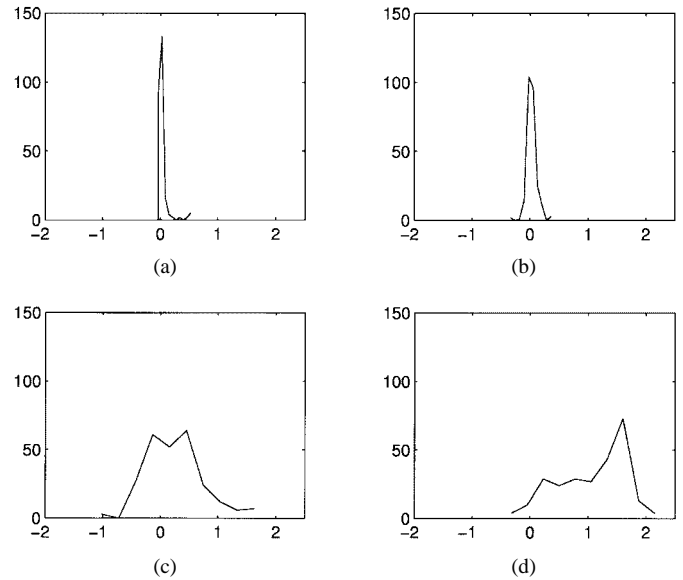


Fig. 9. Distributions of the values of the designed parameters. (a) α_0 . (b) α_1 . (c) α_2 . (d) α_3 .

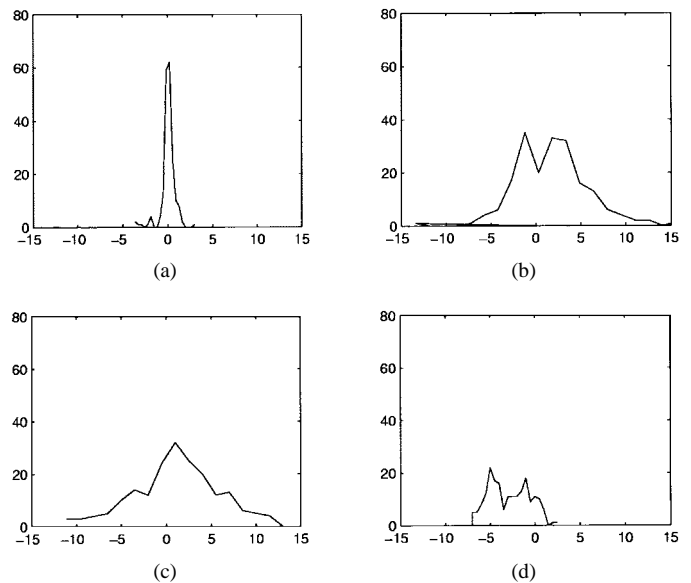


Fig. 10. Distributions of the values of the combination weighting. (a) β_1 . (b) β_2 . (c) β_3 . (d) β_4 .

indicates that a single filter bank is not sufficient to attain the optimum compression. Since the side information to encode the parameters can be kept very small, the adaptive one is a better choice.

Now, we discuss the choice of the weighting parameters. This is the choice of the representation space. The distributions of the values of the design parameters α_i for image ‘‘Lena’’ are shown in Fig. 9. The distributions of α_0 and α_1 are concentrated in the neighborhood of zero. The distributions of α_2 and α_3 are within the interval $[-1, 1]$ and $[0, 2]$, respectively. These distributions of the values of the parameters β_i for image ‘‘Lena’’ are shown in Fig. 10. In this case, the dynamic ranges of β_i are larger than that of α_i . This is because the frequency compensation form provided by the Bernstein filters are more stable than the time domain adaptation form provided by the

TABLE I
ENTROPY OF PREDICTION ERRORS OF TEST IMAGES

Images	MED	GAP	Adaptive IDPCM pyramid
couple	2.74	3.00	2.80
face	4.79	4.65	4.61
girl	4.17	4.08	4.09
hat	4.50	4.41	4.40
jet	3.93	4.25	4.28
lena	4.55	4.39	4.35
london	3.78	3.93	3.77
baboon	5.77	6.26	6.04
MRI1	4.44	4.40	4.24
MRI2	4.74	4.68	4.58
MRI3	4.40	4.34	4.29
MRI4	4.58	4.54	4.53
MRI5	4.45	4.41	4.37
Average	4.37	4.41	4.33

Maxflat filters. From the coding point of view, α_i is chosen rather than β_i .

In the following, the efficiency of some different adaptive prediction schemes are investigated based on the measure of zero-order entropy, as shown in Table I. The first predictor considered is the median edge detector (MED) [14]. The MED is used as the predictor in LOCO-I for lossless image compression [13]. The MED selects the median from a set of three predictions to predict the current pixel. The second predictor considered is the gradient adjusted predictor (GAP) [13]. The GAP is used as the predictor in CALIC [15]. CALIC is a lossless image coder with an extra prediction stage that handles the prediction error. This technique will be briefly reviewed in the following. The GAP adapts the prediction according to local gradients in the neighborhood of the current predicted pixel. One point to be noted is that both the CALIC and LOCO-I are single-resolution predictive coding schemes. The entropy of prediction errors with the MED, GAP, and IDPCM predictor are listed in the second, third, and fourth columns in Table I, respectively. As can be seen from the results, minimizing the prediction error leads to the lowest average entropy over the entire test images.

Now, let us discuss some well known techniques that are used to deal with the prediction error for efficient coding. The performance of a predictor encoder can often be improved with these techniques. In multirate prediction, the zero-tree and set partitioning are very efficient to encode the prediction error. In sequential prediction, errors modeling and feedback technique proposed in [15] perform very well. In the following, we first investigate the performance of error modeling in the multirate prediction. Then, we examine the encoder performance affected by the use of error modeling and set partitioning.

In CALIC, the contexts to model the prediction errors consist of the texture contexts and the energy contexts. Texture contexts are formed by quantizing the local neighborhood pixels into a binary vector. The vector of local values is selected as

$$[x_0, \dots, x_7] = [n, w, nw, ne, nn, ww, 2n - nn, 2w - ww] \quad (40)$$

where $n, w, nw, ne, nn,$ and ww denote north, west, north-west, north-east, north-north, and west-west neighbors, respec-

TABLE II
ENTROPY OF PREDICTION ERRORS OF TEST IMAGES

Images	CALIC	Adaptive IDPCM pyramid + error modeling
couple	2.57	2.79
face	4.58	4.59
girl	3.97	4.08
hat	4.31	4.18
jet	4.17	4.02
lena	4.29	4.25
london	3.74	3.75
baboon	6.16	6.09
MRI1	4.17	4.24
MRI2	4.47	4.43
MRI3	4.15	4.18
MRI4	4.34	4.51
MRI5	4.21	4.32
Average	4.24	4.26

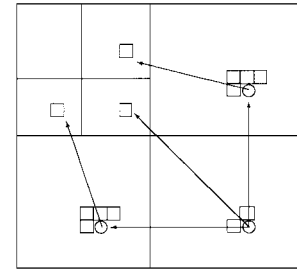


Fig. 11. Contexts used for texture contexts.

TABLE III
AVERAGE ENTROPY AT DIFFERENT RESOLUTION LEVELS

	$\bar{d}^1[k]$	$\bar{d}^2[k]$	$\bar{d}^3[k]$	$\bar{d}^4[k]$
No error modeling	4.18	4.62	5.38	6.00
Have error modeling	4.10	4.61	5.38	6.00

tively. The vector is then quantized to an eight-bit binary number $b_7b_6 \dots b_0$ using the prediction value as the threshold [15]. Energy contexts are formed by quantizing the error energy Δ , which is defined as

$$\Delta = ad_h + bd_v + c|e_w| \quad (41)$$

where $d_h, d_v,$ and e_w are defined in [13, (5)]. We chose $a = b = 1$ and $c = 2$. Δ is then quantized to eight levels by using the bins

$$\begin{aligned} q_1 &= 5, & q_2 &= 15, & q_3 &= 25, & q_4 &= 42 \\ q_5 &= 60, & q_6 &= 85, & q_7 &= 140. \end{aligned} \quad (42)$$

With these settings, the error modeling is done by using 2^8 texture contexts and eight energy contexts for a total of 2048 compound contexts. In CALIC, the estimated error is then compared with the GAP prediction error to generate an improved prediction error. With this approach, the entropy of the prediction errors with different images with CALIC is listed in the second column in Table II. It can be seen that the performance is improved about 3.85% from 4.41–4.24 b/pixel.

The use of error modeling in the multirate prediction has also been mentioned in [31]. The method employed is similar

TABLE IV
ACTUAL BIT-RATE OF TEST IMAGES

Images	S	S+P	Non-Adaptive	Adaptive IDPCM pyramid	+error modeling
couple	3.72	2.93	3.23	3.09	3.08
face	5.14	4.92	4.88	4.75	4.73
girl	4.65	4.42	4.44	4.31	4.30
hat	5.08	4.69	4.73	4.64	4.58
jet	4.85	4.61	4.69	4.58	4.50
lena	4.57	4.43	4.47	4.37	4.31
london	4.33	3.96	4.25	4.14	4.12
baboon	6.66	6.49	6.53	6.42	6.48
MRI1	4.87	4.18	4.58	4.25	4.25
MRI2	5.21	4.58	4.96	4.51	4.48
MRI3	4.86	4.25	4.60	4.20	4.11
MRI4	5.07	4.46	4.89	4.78	4.76
MRI5	4.92	4.32	4.66	4.24	4.19
Average	4.92	4.48	4.69	4.48	4.45

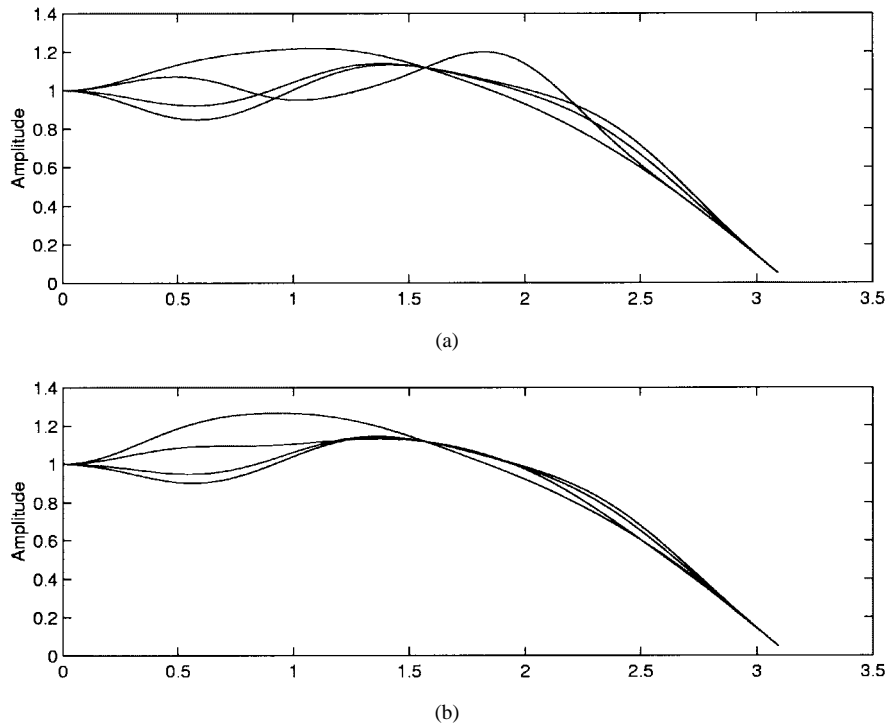


Fig. 12. Frequency responses of the resultant lowpass analysis filters. (a) First case. (b) Second case.

to that used in CALIC. The difference is the use of context information. The contexts used for texture contexts are shown in Fig. 11. For the LH (low-high) band and the HL (high-low) band, the texture contexts are formed by the four neighboring prediction errors and the parent errors. For the HH (high-high) band, the two corresponding coefficients in the LH band and HL band are used instead of the diagonally adjacent neighbors. Prediction errors within the context are then quantized into five levels using ± 3 and ± 13 as threshold. Energy contexts are formed by quantizing the error energy in LH, HL, and HH bands, respectively, as

$$\Delta_{LH} = \delta + a_{lh}|nm| + b_{lh}|p_n| \tag{43}$$

$$\Delta_{HL} = \delta + a_{hl}|ww| + b_{hl}|p_w| \tag{44}$$

$$\Delta_{HH} = \delta + a_{hh}|c_{lh}| + b_{hh}|c_{hl}| \tag{45}$$

where p_n and p_w are the parent prediction errors to the north and west, and c_{lh} and c_{hl} are the brother prediction errors

TABLE V
ENTROPY OF PREDICTION ERRORS OF TEST IMAGES

Images	No primal lifting	With primal lifting 1	With primal lifting 2
couple	2.80	2.87	3.11
face	4.61	4.68	4.70
girl	4.09	4.13	4.08
hat	4.40	4.46	4.49
jet	4.28	4.37	4.52
lena	4.35	4.41	4.44
london	3.77	3.81	4.25
baboon	6.04	6.09	6.23
MRI1	4.24	4.32	4.53
MRI2	4.58	4.63	4.87
MRI3	4.29	4.38	4.48
MRI4	4.53	4.62	4.72
MRI5	4.37	4.43	4.56
Average	4.33	4.40	4.54

of the current pixels, where δ is defined [31, (2)]; we chose $a_{lh}, a_{hl}, a_{hh}, b_{lh}, b_{hl},$ and b_{hh} all equal to 1. Energy contexts

TABLE VI
ACTUAL BIT-RATE OF TEST IMAGES

Images	With primal lifting 1	+error modeling	With primal lifting 2	+error modeling
couple	3.14	3.10	3.24	3.19
face	4.79	4.76	4.93	4.89
girl	4.35	4.33	4.35	4.33
hat	4.70	4.64	4.80	4.73
jet	4.64	4.59	4.68	4.62
lena	4.42	4.37	4.56	4.50
london	4.15	4.12	4.26	4.22
baboon	6.45	6.41	6.58	6.53
MRI1	4.29	4.25	4.36	4.30
MRI2	4.54	4.51	4.59	4.55
MRI3	4.21	4.17	4.26	4.21
MRI4	4.79	4.76	4.84	4.76
MRI5	4.26	4.25	4.40	4.38
Average	4.52	4.48	4.60	4.55

are quantized to eight levels. Then, the error modeling is done by using 5^5 texture contexts and eight energy contexts for a total of 25 000 compound contexts. With the context-based error modeling in the adaptive IDPCM prediction, the entropy of the prediction errors for different images are listed in the third column in Table II. The improvement is about 1.62% from 4.33–4.26 b/pixel.

There can be many reasons that the improvement in the multirate case is less than that in the sequential case. Several possible reasons include

- 1) the setting of the threshold values;
- 2) the selection of context information;
- 3) the neighborhood relationship is weaker in the multirate environment (especially in the lower resolution levels), etc.

The error modeling has no effect in the lower resolution levels. Since most of the large prediction errors are found in these levels, how to utilize the error modeling at these levels becomes an interesting problem. The entropy improvement of the prediction errors at different resolution levels are shown in Table III. It can be seen that the entropy reduction by the error modeling is only significant at the fine resolution level $J = 1$.

Now, let us consider the multirate encoding. The coding results for various images are shown in Table IV. To encode the prediction error, the set partitioning in hierarchical trees (SPIHT) method is used. The test images include the natural images and the medical MRI images. In this simulation, the block size is 32×32 corresponding to four decomposition levels. The order of the halfband filters is four ($N = 4$). From Fig. 9, it can be seen that α_0 is near zero. This indicates that at least two zeros at π is necessary for practical interpolation corresponding to a linear interpolation. The value of α_i can also indicate the regularity of the filter. From the above observation, the α_0 is set to be zero, and the remaining three free parameters $\alpha_1 \sim \alpha_3$ are decided by the adaptive interpolation algorithm. If the length of the filter is $4N - 1$, we need $2N$ neighborhood points at resolution j for prediction a point at resolution $j - 1$. For comparison, the results with the S transform and the S+P transform are also shown in this table. In the S+P transform, the predictor C used in [1] is chosen. The nonadaptive case is also shown. In this case, the seven-tap Maxflat filter is used. In the case of adaptive IDPCM pyramid,

the results with error modeling are also shown. In most cases, the adaptive IDPCM pyramid performs better than the S+P transform. The slight improvement due to the error modeling can also be seen.

Direct decimation causes the aliasing problem. To remove this, the primal lifting step is needed. A primal lifting step consists of applying an update filter $s[l]$ to the highpass components to update the lowpass components. Let us denote $\tilde{c}^j[k]$ as the updated lowpass component. The primal lifting [7] can be described as

$$\tilde{c}^j[k] = c^j[k] + \sum_l \bar{d}^j[k-l]s[l]. \quad (46)$$

With the primal lifting step, the equivalent analysis lowpass filter can be described in the z domain as

$$H(z) = 1 + zP(-z)S(z^2) \quad (47)$$

where $S(z)$ is the z transform of $s[l]$. In general, one lowpass filter can be obtained if the updated filter $S(z)$ is chosen appropriately.

One important thing to note is that the primal lifting step will change the lowpass sequence during the iterative prediction. Thus, not all the decimated pixels are used to determine the α_i . Further, a full adaptive lifting scheme is too complex for the practical applications. Therefore, two alternative approaches are considered in our experiments. The updated filter $s[l]$ used in the primal lifting steps is kept fixed and set to be $s[l] = (\delta[l]/2)$. In the first case, only the pixels in the finest resolution level $J = 1$ are used to determine the α_i values within the block. The frequency responses of the analysis lowpass filters $H(z)$ are shown in Fig. 12(a). It can be seen that effective lowpass filters to reduce the aliasing errors can be derived with the primal lifting steps. The resultant entropy are listed in the third column in Table V. In the second case, the α_i values are determined the same way as the case without the primal lifting steps. The frequency responses of the analysis lowpass filters $H(z)$ are shown in Fig. 12(b). The resultant entropy are listed in the fourth column in Table V. For both of the two approaches, the parameters are not optimal in minimizing the total prediction errors. However, as shown in Table V, for the first case, only a slight degradation is seen. The way to derive the optimal α_i values efficiently when the primal lifting steps

are considered is still under investigation. The corresponding bit rates are shown in Table VI. The average bit rate is about the same as that of S+P and is slightly worse than that without the primal lifting.

V. CONCLUSIONS

A design method for adaptive predictor in the lifting scheme was discussed. Based on the polynomial interpolation, a set of basis filters is proposed to construct the predictor. The nonmaximally flat adaptive predictor is shown to be useful for interpolation of time-varying signals. For efficient compression, the dynamic ranges of the weighting variables are specially considered, and a different representation space is used to represent them. With only the dual lifting, an algorithm has been derived to minimize the total prediction error. With only the dual steps, parallel computation of all the prediction errors can be achieved. However, for progressive transmission, the primal lifting is needed to reduce the aliasing error. This will slightly increase the bit rate. Simultaneous minimization of the prediction error and reduction of the aliasing error is a problem for future research. In this case, an iterative step to determine the predictor parameters is needed.

REFERENCES

- [1] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy image compression," *IEEE Trans. Image Processing*, vol. 5, pp. 1303–1310, 1996.
- [2] A. R. Calderbank, I. Daubechies, W. Sweldens, and B. L. Yeo, "Lossless image compression using integer to integer wavelet transform," in *Proc. IEEE ICIP*, 1997, pp. 596–599.
- [3] H. Y. Jung and R. Prost, "Rounding transform based approach for lossless subband coding," in *Proc. IEEE ICIP*, 1997, pp. 274–277.
- [4] M. Iwahashi, S. Fukuma, and N. Kambayashi, "Lossless coding of still images with four channel prediction," in *Proc. IEEE ICIP*, 1997, pp. 266–269.
- [5] K. Komatsu and K. Sezaki, "Reversible subband coding of image," in *Proc. SPIE VCIP*, 1995, pp. 676–684.
- [6] O. Egger and M. Kunt, "Embedded zerotree based lossless image coding," in *Proc. IEEE ICIP*, 1995, pp. 616–619.
- [7] W. Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, pp. 511–546, 1998.
- [8] ———, "The lifting scheme: A new philosophy in biorthogonal wavelet construction," *Proc. SPIE*, pp. 68–79, 1995.
- [9] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Cambridge, MA: Wellesley-Cambridge, 1996.
- [10] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992.
- [11] O. Herrmann, "On the approximation problem in nonrecursive digital filter design," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 411–413, 1971.
- [12] L. R. Rajagopal and S. C. Dutta-Roy, "Design of maximally-flat FIR filters using the bernstein polynomial," *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 1587–1590, 1987.
- [13] N. Memon, V. Sippy, and X. Wu, "A comparison of prediction scheme proposed for a new lossless image compression standard," in *Proc. IEEE ISCAS*, 1996, pp. 309–312.
- [14] S. A. Martucci, "Reversible compression of HDTV images using median adaptive prediction and arithmetic coding," in *Proc. IEEE ISCAS*, 1990, pp. 1301–1313.
- [15] X. Wu, "Lossless compression of continuous-tone images via context selection, quantization, and modeling," *IEEE Trans. Image Processing*, vol. 7, pp. 656–664, 1997.
- [16] D. Donoho, "Interpolating wavelet transform," Dept. Statist., Stanford Univ., Stanford, CA, preprint.
- [17] B. R. Hunt, "An optical analogy to DPCM digital image data compression," *SPIE Appl. Digital Image Process.*, pp. 85–95, 1977.
- [18] ———, "Optical computing for image bandwidth compression: analysis and simulation," *Appl. Opt.*, pp. 2944–2951, 1978.
- [19] L. Arnold, "Interpolative coding of image with temporally increasing resolution," *Signal Process.*, pp. 151–160, 1989.
- [20] T. Endoh and Y. Yamazaki, "Progressive coding scheme for multilevel images," in *Proc. Picture Coding Symp.*, 1986, pp. 21–22.
- [21] P. Roos, A. Viergever, M. C. A. Van Dijke, and J. H. Peters, "Reversible intraframe compression of medical images," *IEEE Trans. Med. Imag.*, vol. 7, pp. 328–336, 1988.
- [22] E. A. Gifford, B. R. Hunt, and M. W. Marcellin, "Image coding using adaptive recursive interpolation DPCM," *IEEE Trans. Image Processing*, vol. 4, pp. 1061–1069, 1995.
- [23] M. Vetterli and C. Herley, "Wavelets and filter banks: Theory and design," *IEEE Trans. Signal Processing*, vol. 40, pp. 2207–2232, 1992.
- [24] R. A. Gopinath and C. S. Burrus, "Wavelet-based lowpass/bandpass interpolation," in *Proc. IEEE ICASSP*, 1992, pp. 385–388.
- [25] N. Saito and G. Beylkin, "Multiresolution representations using the auto-correlation functions of compactly supported wavelets" in *Proc. IEEE ICASSP*, 1992, pp. 381–384.
- [26] S. Ries, "On the reconstruction of signal by a finite number of samples," *Signal Process.*, pp. 45–68, 1991.
- [27] H. Caglar and A. N. Akansu, "A generalized parametric PR-QMF design technique based on Bernstein polynomial approximation," *IEEE Trans. Signal Processing*, vol. 41, pp. 2314–2321, 1993.
- [28] T. Cooklev, A. Nishihara, and M. Sablatash, "Regular orthonormal and biorthogonal wavelet filters," *Signal Process.*, pp. 121–137, 1997.
- [29] Farin and Gerald, *Curve and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Orlando, FL: Academic, 1988.
- [30] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.
- [31] N. Memon, X. Wu, and B.-L. Yeo, "Improved techniques for lossless image compression with reversible integer wavelet transforms," in *Proc. IEEE ICIP*, 1998, pp. 891–895.



Wen-Jen Ho received the B.S. degree in electrical engineering from the National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., and the M.S. degree in telecommunication engineering and the Ph.D. degree in electronic engineering, both from the National Chiao Tung University, Hsinchu, Taiwan, in 1991 and 1998, respectively.

He is presently a Senior Engineer with the Institute for Information Industry in Taiwan. His research interests are image processing, video coding and video object segmentation.



Wen-Thong Chang was born in Taiwan, R.O.C., in 1956. He received the B.S. degree from National Chiao Tung University, Hsinchu, Taiwan, in 1979, the M.S. degree from the State University of New York, Stony Brook, in 1981, and the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 1985.

He is now a Professor in the Department of Communication Engineering, National Chiao Tung University. His research interests include digital signal processing, digital image processing, video compression, and communication system.