# Searches for optimal second generation wavelet bases for use in signal coding

M. Howlett and D. T. Nguyen

***Abstract*— We use the interpolating formulation of the lifting scheme to construct optimized second–generation wavelets for a given segment of data. We optimize with respect to an energy compaction cost function - the Shannon entropy of the uniformly quantized transform coefficients. We determine the *predict* operator adaptively at each level (and thus there is no dilation relation between wavelets on different scales) and calculate an associated *update* operator to produce vanishing moments in the synthesis wavelet. Our initial adaptive algorithm is an inefficient search over the space of 4 tap linear predictor operators sampled at a finite resolution. This search provides a good estimate of the best we can hope to achieve using any efficient algorithm for the same task. We present results for three typical segments of PCM audio. A further 0 - 5.6% reduction in Shannon entropy (an improvement of 0 - 22%) is seen when compared to the standard polynomial interpolation predictors in our examples. Finally, we discuss possible techniques for determining an optimal, or near optimal linear predict operator efficiently for a given signal.**

***Index Terms*—signal compression, second–generation wavelets, lifting scheme, adaptive wavelets**

## I. Introduction

THE first stage in the majority of standard lossy signal compression algorithms is a linear transform. The primary purpose of this transform is to convert the original signal (PCM audio, image etc.) into a representation in which there are as few significant coefficients as possible. The de–correlated coefficients can then be more efficiently coded. Recently, wavelet transforms have been replacing the discrete cosine transform as the transform of choice in coding applications. They have been formally adopted into the MPEG-4 standard for the compression of background information in video; in the JPEG200 standard for the compression of still images; and in the FBI standard for the compression of fingerprints. The wavelets used in each of these standards are traditional "first–generation wavelets", that is, the basis functions are all translates and dilates of each other. Recently, Sweldens [2] proposed a more general framework in which the translation / dilation requirement is relaxed. The trick, of course, is to find bases in this new "second generation" setting that out perform the most effective first generation bases. In this paper, we use the interpolating formulation of the lifting scheme to construct second–generation wavelets optimized for a given segment of data and compare the results to the non–adaptive first generation wavelets in the same class.

Matt Howlett is a PhD candidate at the School of Engineering, University of Tasmania, Hobart, Tasmania, Australia, 7000. He can be contacted via email: mhowlett@utas.edu.au, Tel: (03) 62261857, and Fax: (03) 62262136

Thong Nguyen is professor of Electrical Engineering at the School of Engineering, University of Tasmania (email: D.T.Nguyen@utas.edu.au).

ISBN: 1-86467-114-9

We consider one dimensional transforms only, but the results generalise to higher dimensions.

In section II, we review filter banks, the lifting scheme and the discrete wavelet transform. We also introduce the predict and update lifting operators. In section III we provide details of the method used to search for the optimal linear predict operator and in section IV we detail the measure used to assess the filters. In section V we present results for three typical PCM audio signals. Finally, in section VI, we comment on potential efficient strategies for determining optimal filters.

## II. The lifting scheme

The *lifting scheme* [2][3] is a technique for constructing a set of biorthogonal filters by modifying another existing set. Usually, one starts with a trivial set of filters and applies a series of lifting and dual lifting steps to create a new set of filters with desirable properties.

Biorthogonal filter banks give rise to biorthognoal wavelets via the discrete wavelet transform (DWT). The DWT is simply the process of applying a two channel filter bank (with down sampling) iteratively to the low pass sub–band (initially the original signal) and saving the values in high pass sub–band at each step, as well as the final coarse approximation. These values correspond to coefficients of wavelets at increasingly large scales, and smoothing functions at the lowest scale. Together these functions form a basis for the the space of input signals. For a detailed introduction to filter banks and wavelet theory, the reader is referred to the text by Vetterli and Kovacevic [5].

In the interpolating formulation of lifting (see figure 1), one starts with a polyphase decomposition which simply splits the input signal into odd and even components. Next, a dual lifting step is applied. This corresponds to predicting the odd components using the even components and storing the difference. If the signal is relatively smooth, the odd components will be close to their predicted values and the stored values will be small. On the other hand, where the signal has significant fine detail, it will not be possible to predict the odd values very well and the differences will be large. Thus the stored values correspond to the detail or high frequency information in the signal. In $z$–transform notation, the detail signal (high pass sub–band) is given by:

$$HP(z) = x_o(z) - T(z)x_e(z) \tag{1}$$

where $T(z)$ is the predict operator and $x_e(z)$ and $x_o(z)$ are the odd and even signal components respectively. A simple example of a predict operator is:

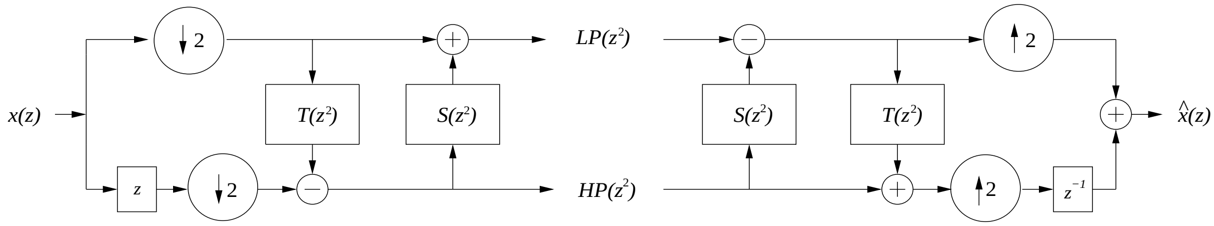


Fig. 1. A block diagram showing the construction of the low and high pass sub–bands and subsequent reconstruction of the original signal. First, a dual lifting step (the predict operator) is applied, then a lifting step (the update operator). The signal is reconstructed by simply un–doing each operation in the forward process, but in reverse order.

$$T(z) = \frac{1}{2}z + \frac{1}{2} \tag{2}$$

This corresponds to predicting each odd value as an average of its two neighbouring even values. In other words, the detail (high pass) coefficients are the failure of the signal to be piecewise linear. Neville's algorithm (see for example [6]) can be used to compute filters that correspond to higher order interpolation.

After constructing the high–pass subband, the even values are *updated* to produce a coarse approximation of the original signal. In fact the even components already constitute a coarse approximation without any modification, however it is not a very good one - simply sub–sampling a signal introduces considerable aliasing. It is possible to use the newly constructed detail coefficients to *update* the even signal values so as to create a better coarse representation. The coarse approximation (lowpass sub–band) is given by

$$LP(z) = x_e(z) + S(z)HP(z) \tag{3}$$

where $S(z)$ is the update operator. The update operator $S(z)$ associated with a given predict operator $T(z)$ is usually chosen to conserve some number of moments in the low pass (coarse) sub–band. This is equivalent to ensuring the associated high pass synthesis filter has the same number of vanishing moments. An update operator that complements the linear predict operator in equation 2 is given by:

$$S(z) = \frac{1}{4} + \frac{1}{4}z^{-1} \tag{4}$$

A non trivial calculation is required to obtain this result. In general, the update operator may be determined from a linear system of equations in which the variables are the update filter coefficients, and each additional constraint on the moments of the high–pass synthesis filter adds an additional equation to the linear system [3].

One very appealing side effect of implementing an analysis filter bank using lifting steps is that the inverse process is immediately apparent. First one un–does the effect of the update operator to reconstruct the even values as follows:

$$x_e(z) = LP(z) - S(z)HP(z) \tag{5}$$

The odd coefficients can then be recovered by reversing the predict step:

$$x_o(z) = HP(z) + T(z)x_e(z). \tag{6}$$

The process of applying the predict operator followed by the update operator can be written in terms of polyphase matrices:

$$\begin{aligned} \begin{bmatrix} LP(z) \\ HP(z) \end{bmatrix} &= \begin{bmatrix} 1 & S(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -T(z) & 1 \end{bmatrix} \begin{bmatrix} x_e(z) \\ x_o(z) \end{bmatrix} \\ &= \begin{bmatrix} 1 - S(z)T(z) & S(z) \\ -T(z) & 1 \end{bmatrix} \begin{bmatrix} x_e(z) \\ x_o(z) \end{bmatrix} \end{aligned}$$

The inverse process can be written:

$$\begin{aligned} \begin{bmatrix} x_e(z) \\ x_o(z) \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ T(z) & 1 \end{bmatrix} \begin{bmatrix} 1 & -S(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} LP(z) \\ HP(z) \end{bmatrix} \\ &= \begin{bmatrix} 1 & -S(z) \\ T(z) & 1 - T(z)S(z) \end{bmatrix} \begin{bmatrix} LP(z) \\ HP(z) \end{bmatrix} \end{aligned}$$

The high and low pass analysis filters ($\widetilde{G}(z^{-1})$, $\widetilde{H}(z^{-1})$) and the high and low pass synthesis filters ($G(z)$, $H(z)$) can be obtained directly from the polyphase matrices [5]:

$$\widetilde{G}(z^{-1}) = z - T(z^2) \tag{7}$$

$$\widetilde{H}(z^{-1}) = 1 - S(z^2)T(z^2) + zS(z^2) \tag{8}$$

$$G(z) = -S(z) + z^{-1}\left[1 - S(z^2)T(z^2)\right] \tag{9}$$

$$H(z) = 1 + z^{-1}T(z^2) \tag{10}$$

## III. A Selective Search

The premise of the current work is that the optimal linear predictor for a *given* segment of data will not, in general, be a polynomial interpolation operator. For smooth signals, the optimal predictor of length $N$ will be very close to the the polynomial interpolation operator or order $N$. However, for signals with a significant high frequency (fine detail) component, the optimal filter is likely to be very different.

To attempt to determine the optimal prediction filter we use a guided search. This method is not useful in any practical situation because it is highly inefficient. However, it allows us to estimate an upper limit on what we can hope to achieve using *any* efficient method. This is important for two reasons. Firstly, it tells us if it really is worth our while trying to devise efficient adaptive linear predictor algorithms. If we can typically only achieve a marginal improvement, then the small gain is probably not worth the effort and added computational cost. Secondly, assuming that we determine that it is worth our while, the current work allows us to estimate how close to optimal our yet–to–be–devised efficient algorithms actually are.

For simplicity, consider searching the space of all high–pass analysis filters $\widetilde{G}(z^{-1})$ that may be constructed by dual lifting with a linear filter $T(z)$ with two taps only. The following discussion easily generalises to longer predict operators. Let the predict operator be given by:

$$T(z) = t_0 z + t_1$$

The filter $\widetilde{G}(z^{-1})$ then takes the form:

$$\widetilde{G}(z^{-1}) = -t_0 z^2 + z - t_1 \qquad (11)$$

For the purposes of coding, the normalization of the transform coefficients is irrelevant. Thus, due to linearity, the normalization of the filter bank filters (and in particular $\widetilde{G}(z^{-1})$) is also irrelevant. Therefore, we may consider analysis high–pass filters of the form:

$$\widetilde{G}(z^{-1}) = g_0 z^2 + g_1 z + g_2$$

where $g_0, g_1, g_2 \in \mathcal{R}$ and understand that the predict operator obtained by setting:

$$t_0 = \frac{-g_0}{g_1}, \quad t_1 = \frac{-g_2}{g_1} \qquad (12)$$

produces transform coefficients that are equally codable.

We cannot search every combination of $g_0, g_1$ and $g_2$ since there are infinitely many of them. We can perform an exhaustive search at a given "resolution" however. The resolution of the search space is determined by the smallest possible *ratio* between filter coefficients. We search

$$-2^R \leq g_0, g_2 \leq 2^R, \;\; 1 \leq g_1 \leq 2^R$$

where $R$ denotes the "resolution level" and $g_0, g_1$ and $g_2$ are integers. At at level $R$, the magnitude of the smallest possible ratio between two coefficients is $r = 2^{-R}$. By restricting $g_1$ to be greater than zero, we avoid redundantly considering both $\widetilde{G}(z^{-1})$ and $-\widetilde{G}(z^{-1})$ and ensure that equations 12 are defined. Further redundancy is avoided by only considering cases where the highest common factor of $g_0, g_1$ and $g_2$ is one.

We reduce the search space even further. We are only interested in considering "high–pass" like filters. Traditionally, wavelet basis functions and their associated filters are restricted to have zero mean. Therefore, from equation 11, we must have $t_0 + t_1 = 1$. In fact, we are slightly less restrictive and only require $0.9 < t_0 + t_1 < 1.1$. Unfortunately, this simple restriction does not suffice because although the frequency response $\widetilde{G}(\omega)$ may be (near) zero at $\omega = 0$, there are values of $t_0$ and $t_1$ which cause $\widetilde{G}(\omega)$ to have very strong, relatively low bandpass behavior. To avoid considering these cases, we set a further condition that the frequency spectrum must not reach a magnitude more than twice that of $\widetilde{G}(\pi)$.

Despite all of the above restrictions, an increase by one resolution level still corresponds to an additional factor of $2^N$ in computational time. Thus, the search quickly becomes unmanageable. Therefore, we modify our search as follows. To begin with, we perform a search of the entire space to some manageable resolution $R_B$. We select the $K$ best predictor filters at this resolution and assume that the best filters on the next finest resolution will either be these filters themselves, or nearby filters. We search filters on the next level with coefficients given by

$$g_i' = 2g_i + \delta_{g_i} \qquad (13)$$

where $\delta_{g_i} = -1, 0$ or 1 and $\{g_i\}$ are the coefficients of a filter in the selected set. We then select the $K$ best predict operators at this level and iterate down to some level $R_F$. In the 4 tap case, setting $K \sim 50$ is computationally manageable, whilst still providing a very generous set of filters to check at each level.

Although its construction suggests that this method will find solutions which are close to optimal, there is no guarantee it will converge to the optimal solution.

## IV. Measuring Filter Performance

We determine the performance of a filter over a given segment of data by measuring the energy compaction of the resulting transform coefficients. In their best basis algorithm, Coifman and Wickerhauser [1] use the entropy of the normalised transform coefficients to measure energy compaction. In our fully bi–orthogonal setting, this measure is no longer necessarily appropriate. Indeed, we have tested this measure in our search algorithm and find that it does not produce good results in terms of our own measure.

We use a measure that is directly related to simple variable rate coding algorithms. Firstly, we uniformly quantize the transform coefficients. The optimal bit allocation for bi–orthogonal wavelet coding is discussed in detail by Usevitch [4] and we use this result to determine appropriate quantization step sizes. The expected error in the reconstructed signal is held constant over the entire search.

As our energy compaction measure, we then calculate the minimum expected number of bits to encode the transform coefficients using the Shannon entropy:

$$\mathcal{C} = -\sum_i p_i \log_2 p_i \qquad (14)$$

where $p_i$ is the probability of the $i$th quantized value, estimated empirically from the normalized histogram of quantized coefficients.

## V. Results

We present results for three segments of PCM audio data of length 1024 samples. The sampling rate was 22 kHz in each case. We have chosen to only investigate segments that contain significant mid to high frequency components because we know the polynomial based filters will be very close to optimal for smooth signals. The results presented are typical of other similarly chosen signal segments.

The first segment is a sample from a synthesizer (figure 2). The first level detail coefficients obtained using the polynomial and optimal predict operators are shown in figures 3 and 4 respectively. The second segment is a male voice in the middle of saying the word "I" as in "I went to the shop". The third segment is a snare drum immediately after the attack transient. In each case we only considered predict operators of length 4, and searched down to a resolution level $R_F = 6$ (with $R_B = 2$). We consider transforms of both 1 and 4 levels. For a 1 level adaptive transform, $4 \times 6 + 5 = 29$ additional bits are required at the given resolution to store the predict filter. For a 4 level transform $29 * 4 = 116$ extra bits are required to store the 4 predictors. The expected number of bits to code each segment with an expected reconstructed MSE of $10^{-7}$ are shown in the table below. The corresponding percentage decrease from no transform at all is shown in parentheses. All results include the cost of storing the predict operator coefficients where necessary.

| transform | synth | voice | snare |
|---|---|---|---|
| none | 8136 | 9414 | 9077 |
| poly (1) | 6610 (18.7) | 7344 (21.5) | 7198 (20.7) |
| optimal (1) | 6267 (23.0) | 6932 (26.4) | 7212 (20.5) |
| poly (4) | 5928 (27.1) | 6433 (31.7) | 6280 (30.8) |
| optimal (4) | 5479 (32.7) | 6078 (35.4) | 6258 (31.1) |

For both the synthesizer and voice segments we see a significant decrease in expected coding size when using the optimal predict operators over of the polynomial interpolation ones. From the single level transform of the synthesizer segment, we see the improvement is as high as 22%.

Despite containing significant detail on all scales, we see very little difference between the expected coded sizes for the snare drum segment after application of optimal and polynomial filters. We postulate that this because the signal is highly non–stationary

## VI. Efficient Algorithms

As a first step toward developing an efficient algorithm for determining the predict operator adaptively, we use existing adaptive filter theory and set the even samples to be the input signal and the odd samples the desired response. Our initial algorithms choose the predict operator for a given section of data as the average, or median LMS or RLS adaptive filter over the same region. Despite the fact that these adaptive algorithms are designed for the case where the filter is slowly changing (in our setting this is not generally true) we have found that in some situations the derived filters perform significantly better than the polynomial interpolation filters of the same length. In most situations however, the performance is worse.

## VII. Conclusion

When considering a variety of non–smooth signals, we have demonstrated that a significant improvement in coding gain (typically a further 0 - 6%) is possible by choosing predict operators adaptively. Current work is concerned with designing an effective and efficient adaptive algorithm for this purpose.

## References


[1] Ronald R. Coifman and Mladen Victor Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38(2):713–718, 1992.

[2] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.*, 3(2):186–200, 1996.

[3] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1997.

[4] B. Usevitch. Optimal bit allocation for biorthogonal wavelet coding. In *Proc. Data Compression Conference*, pages 387–395, Snowbird, Utah, March 1996.

[5] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice-Hall, 1995.

[6] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes*. Cambridge University Press, 1992.


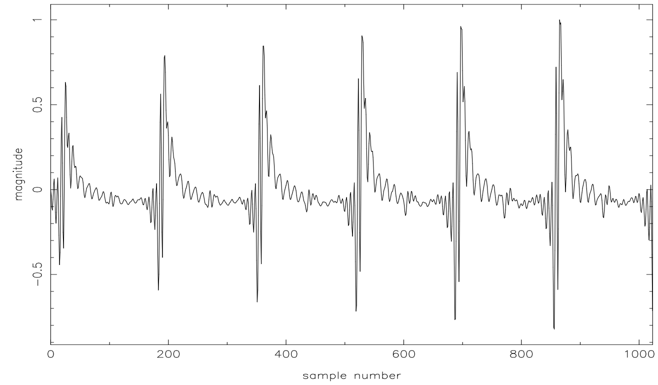


Fig. 2. The synthesizer data segment.

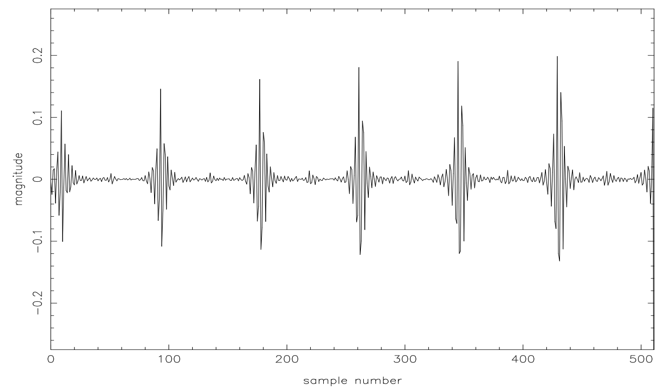


Fig. 3. First level detail coefficients of the synthesizer data after application of the polynomial interpolation filter. A contribution to the output MSE of $0.5 \times 10^{-7}$ corresponds to a quantization step size of $1.29 \times 10^{-3}$.

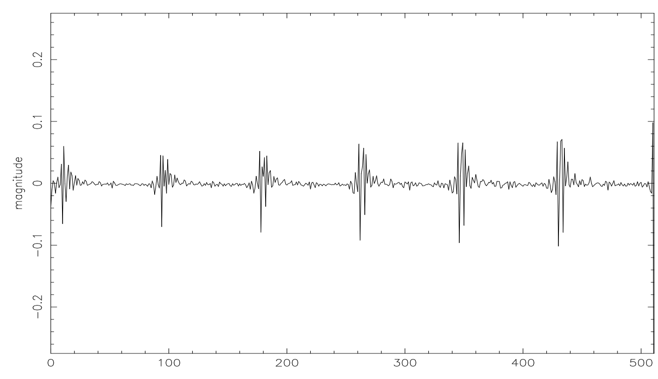


Fig. 4. First level detail coefficients of the synthesizer data after application of the optimal filter. A contribution to the output MSE of $0.5 \times 10^{-7}$ corresponds to a quantization step size of $1.36 \times 10^{-3}$.