ADAPTIVE WAVELET TRANSFORMS VIA LIFTING

Roger L. Claypoole, Jr. and Richard G. Baraniuk

Dept. of Electrical and Computer Engineering
Rice University
Houston, TX 77005

Robert D. Nowak

Dept. of Electrical Engineering
Michigan State University
East Lansing, MI 48824

ABSTRACT

This paper develops two new adaptive wavelet transforms based on the lifting scheme. The lifting construction exploits a spatial-domain, prediction-error interpretation of the wavelet transform and provides a powerful framework for designing customized transforms. We use the lifting construction to adaptively tune a wavelet transform to a desired signal by optimizing data-based prediction error criteria. The performances of the new transforms are compared to existing wavelet transforms, and applications to signal denoising are investigated.

1. INTRODUCTION

The discrete wavelet transform (DWT) provides a very efficient representation for a broad range of real-world signals. This property has been exploited to develop powerful signal denoising and estimation methods [1] and extremely low-bit-rate compression algorithms [2].

The 1-d DWT represents a real-valued discrete-time signal in terms of shifts and dilations of a lowpass scaling function and a bandpass wavelet function [2]. The DWT decomposition is multiscale: it consists of a set of *scaling coefficients* $c^0[n]$, which represent coarse signal information at scale $j = 0$, and a set of *wavelet coefficients* $d^j[n]$, which represent detail information at scales $j = 1, 2, \ldots, J$. The forward DWT has an efficient implementation in terms of a recursive multirate filterbank based around a lowpass filter h and highpass filter g. The inverse DWT employs an inverse filterbank with lowpass filter $\widetilde{h}$ and highpass filter $\widetilde{g}$. For special choices of h and g, we have $\widetilde{h} = h$ and $\widetilde{g} = g$, and the underlying wavelet and scaling and wavelet functions form an orthonormal signal basis. Otherwise, these functions form a biorthogonal basis [2].

The economy of the wavelet transform stems from the fact that the DWT tends to compress real-world signals into just a few coefficients of large magnitude. Compression follow from the "vanishing moments" property of wavelets, which guarantees that the wavelet coefficients of low-order polynomial signals are zero [2]. Thus, if a signal is exactly polynomial, then it can be completely described using scaling coefficients alone. In more realistic situations, the signal will not be polynomial, but may be well-approximated by a piecewise polynomial function. Because wavelet functions also have localized support, most of the wavelet coefficients of such a signal will be zero except those corresponding to wavelets having support near the breakpoints of the polynomial segments.

It is fruitful to view the DWT as a prediction-error decomposition. The scaling coefficients at a given scale (j) are "predictors" for the data at the next higher resolution or scale ($j - 1$). The wavelet coefficients are simply the "prediction errors" between the scaling coefficients and the higher resolution data that they are attempting predict. This interpretation has led to a new framework for DWT design known as the *lifting scheme* [3].

In this paper we use lifting to design customized DWTs that adapt to match the signal under consideration. We develop two new multiscale analysis techniques — *scale-adapted transforms* and *space-adapted transforms*. The fundamental idea in both cases is to adapt the prediction to minimize a data-based error criterion. While other adaptive transform techniques have been proposed in the literature [4, 5, 6], the adaptive transforms developed here are new, particularly in their use of the lifting programme.

The paper is organized as follows. In Section 2, we review the basic lifting construction and describe a variant of the basic scheme. In Section 3, we develop the two new adaptive DWTs using the lifting construction. In Section 4, we apply the new DWTs to signal denoising and demonstrate that the adapted DWTs can perform significantly better than standard wavelet denoising methods in several interesting cases. We close in Section 5 with concluding remarks and plans for future work.

2. THE LIFTING CONCEPT

Lifting, a space-domain construction of biorthogonal wavelets developed by Sweldens [3], consists of the iteration of the following three basic operations (see Figure 1):

Split: Divide the original data into two disjoint subsets. For example, we will split the original data set $x[n]$ into $x_e[n] = x[2n]$, the even indexed points, and $x_o[n] = x[2n+1]$, the odd indexed points.

Predict: Generate the wavelet coefficients $d[n]$ as the error in predicting $x_o[n]$ from $x_e[n]$ using prediction operator $\mathcal{P}$:

$$d[n] \;=\; x_o[n] - \mathcal{P}(x_e[n]). \tag{1}$$

Update: Combine $x_e[n]$ and $d[n]$ to obtain scaling coefficients $c[n]$ that represent a coarse approximation to the original signal $x[n]$. This is accomplished by applying an update operator $\mathcal{U}$ to the wavelet coefficients and adding to $x_e[n]$:

$$c[n] \;=\; x_e[n] + \mathcal{U}(d[n]), \tag{2}$$

Supported by NSF, grant nos. MIP–9457438 and MIP–9701692, ONR, grant no. N00014–95–1–0849, and DARPA through AFOSR grant no. F49620-97-1-0513.

Email: clayporl@rice.edu, richb@rice.edu, nowak@egr.msu.edu
Web: http://www.dsp.rice.edu, http://www.egr.msu.edu/spc/

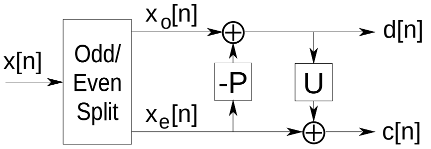

Figure 1: *Lifting stage: Split, Predict, Update.*

These three steps form a *lifting stage*. Iteration of the lifting stage on the output $c[n]$ creates the complete set of DWT scaling and wavelet coefficients $c^j[n]$ and $d^j[n]$.[1]

The lifting steps are easily inverted, *even if $\mathcal{P}$ and $\mathcal{U}$ are non-linear or space-varying*. Rearranging (1) and (2), we have

$$x_e[n] = c[n] - \mathcal{U}(d[n]), \qquad x_o[n] = d[n] + \mathcal{P}(x_e[n]). \tag{3}$$

2.1. Predictor Design

Typically, the prediction operator $\mathcal{P}$ is a linear shift-invariant filter, with z transform $P(z)$. In Figure 2, we illustrate a symmetric, $N = 4$-point predictor $P(z) = p_1 z^{-1} + p_2 + p_3 z + p_4 z^2$. By tracing the contribution of $x_e[n]$ and $x_o[n]$ through the tree to the point $d[n]$, we can find the equivalent filter that would be applied to the original data $x[n]$. In vector form, we have

$$\mathrm{g} \;=\; [-p_1,\ 0,\ -p_2,\ 1,\ -p_3,\ 0,\ -p_4]^T. \tag{4}$$

(Note the zeros at the positions corresponding to odd points in the original data, except for the 1 in the center.)

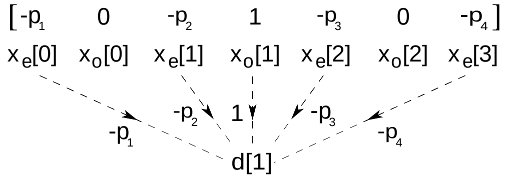

Figure 2: *Prediction filtering. An $N = 4$ point linear prediction filter $P(z)$ yields the prediction vector g shown across the top.*

Recall that the goal of the prediction step is to eliminate all low-order polynomials from $x[n]$ in creating the wavelet coefficients. For a linear predictor, this is easily accomplished by the following simple procedure. Form the $N \times (2N-1)$ matrix $\mathbf{V}$ whose rows are the monomial signals: $[\mathbf{V}]_{m,n} = n^m$, $n = -(N-1), \ldots, (N-1)$, $m = 0, 1, \ldots, N-1$. (We make the convention $0^0 = 1$.) Now, for the predictor to suppress all polynomials up to $N-1$st order, we require that

$$\mathbf{V}\mathrm{g} \;=\; \mathbf{0}. \tag{5}$$

This set of linear equations is readily solved, since $\mathbf{V}$ forms the first N rows of a Vandermonde matrix, which is always invertible [8]. Upon recognizing that the solution lives in an N-dimensional subspace, we can rewrite (5) in a simpler form in terms of a new $N \times N$ matrix $\mathbf{V}^\diamond$ and p the vector of coefficients of the prediction filter $P(z)$

$$\mathbf{V}^\diamond \mathrm{p} \;=\; [1, 0, \ldots, 0]^T. \tag{6}$$

2.2. Update Design

The (linear) update filter $U(z)$ creates $c[n]$ by updating each $x_e[n]$ with the nearest $\widetilde{N}$ wavelet coefficients $d[n]$ from either side. The update order $\widetilde{N}$ can be chosen independently of N; however, the prediction coefficients p_k must be fixed prior to determining the update filter in the standard lifting programme.

In Figure 3, we trace the contribution of the original $x_e[n]$ and $x_o[n]$ to each $c[n]$ for an $N = 2$ point predict followed by an $\widetilde{N} = 4$ point update with $U(z) = u_1 z^{-2} + u_2 z^{-1} + u_3 + u_4 z$. In vector form, we have the equivalent filter h at the top of the Figure. Note that h is a function of both the update coefficients u_k and the prediction coefficients p_k.

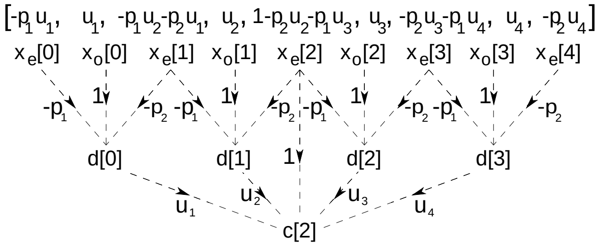
[-p1u1, u1, -p1u2-p2u1, u2, 1-p2u2-p1u3, u3, -p2u3-p1u4, u4, -p2u4]
x_e[0] x_o[0] x_e[1] x_o[1] x_e[2] x_o[2] x_e[3] x_o[3] x_e[4]
-p1
1
-p2
-p1
1
-p2
-p1
1
-p2
-p1
1
-p2
d[0]
d[1]
1
d[2]
d[3]
u1
u2
u3
u4
c[2]

Figure 3: *Update filtering. An $N = 2$ point linear predict followed by an $\widetilde{N} = 4$ point linear update yields the update vector h shown across the top.*

The update filter vector h should pass low-order polynomials into $c[n]$ while attenuating high-order polynomials. Conversely, we can design the mirror update filter vector $\widetilde{\mathrm{g}}$ (defined as $\widetilde{g}_n = (-1)^n h_n$) to *suppress* low-order polynomials. For the example in Figure 3, we have

$$\begin{aligned}\widetilde{\mathrm{g}} \;=\; & [-p_1 u_1,\ -u_1,\ (-p_1 u_2 - p_2 u_1),\ -u_2,\ (1 - p_2 u_2 - p_1 u_3), \\ & -u_3,\ (-p_2 u_3 - p_1 u_4),\ -u_4,\ -p_2 u_4]^T.\end{aligned} \tag{7}$$

Since the $N = 2$ prediction coefficients are already determined, there are $\widetilde{N} = 4$ unknowns (the update coefficients u_k) in $\widetilde{\mathrm{g}}$. Solution of $\mathbf{V}\widetilde{\mathrm{g}} = \mathbf{0}$ as in (5) yields the update coefficients.

In summary, we design the prediction step to eliminate the low-order polynomial signal structure, leaving only the high-order details. We design the update to preserve the low-order polynomial signal structure at the next coarser scale.[2]

2.3. The Update/Predict Programme

In the lifting framework of Figure 1, the update structure depends on the predictor structure. Hence, if $\mathcal{P}$ is space-varying or nonlinear, then so is $\mathcal{U}$, and the design procedure of Section 2.2 becomes unwieldy. A crafty detour around this problem is to perform the update step first, followed by the prediction [9]. The relevant equations then become

$$c[n] = x_e[n] + \mathcal{U}(x_o[n]), \qquad d[n] = c[n] - \mathcal{P}(x_e[n]). \tag{8}$$

After designing a linear update filter to preserve the first $\widetilde{N}$ low-order polynomials in the data (as in Section 2.1), we can apply any

[1] In fact, *all* wavelet transforms can be factored into a series of lifting stages (with perhaps multiple predicts and updates per stage) [7].

[2] In order to normalize the energy of the underlying scaling and wavelet functions, we actually output $2^{-1/2} d[n]$ and $2^{1/2} c[n]$ from the lifting stage.

space-varying or nonlinear predictor without affecting the coarse approximation $c[n]$.

Since the update/predict lifting stage creates $c[n]$ prior to $d[n]$, the prediction operator can be designed to optimize performance criteria other than polynomial suppression capability. For example, the predictor could be a median filter. In Section 3.2, we will exploit this flexibility to design space-varying predictors that adapt to the characteristic of the signal.

3. ADAPTIVE LIFTING

The lifting approach to wavelet design gives us a great deal of flexibility. In principle, we can use any linear, nonlinear, or space-varying predictor and update, and the lifting construction ensures that the resulting transform is invertible. We now investigate the capabilities of the lifting approach for adaptive lifting DWTs that optimize data-based prediction measures to match the characteristics of a given signal. The motivation behind these new transforms is that better predictors will lead to more efficient signal representations. Since the compression abilities of signal transformations are key to successful signal processing algorithms [1], the adaptive transforms derived here have the potential to improve transform-based processing.

3.1. Scale-Adaptive Transforms

In Section 2, we derived the lifting construction based on a polynomial signal suppression/preservation argument. However, we alluded to nonlinear schemes based on other than polynomial prediction. For example, we could design a predictor for certain textural components, such as periodic patterns. More generally, we can let the signal itself dictate the structure of the predictor.

In a scale-adapted transform (ScAT), we adapt the predictor in each lifting stage in order to match signal structure at the corresponding scale. The basic idea is to use a linear N-point predictor, but require that it suppress polynomials only up to $M < N$th order. The remaining $N - M$ degrees of freedom are then used to adapt the predictor to the signal. Specifically, at each scale we optimize the predictor over the $N - M$ degrees of freedom to minimize the spatially-averaged squared prediction error. This optimization produces predictors that can match both polynomial and non-polynomial signal structure. For example, if the signal contains a regular texture, then a relatively low-order adaptive predictor of this form may be able to match the texture much better than a pure polynomial predictor of the same order.

The optimization itself is a straightforward N-dimensional constrained least squares problem — the constraint being that we require the predictor to suppress $M < N$th-order polynomials. Let $\mathbf{x}_\mathrm{o}$ denote the odd-indexed data we wish to predict and let $\mathbf{X}_\mathrm{e}$, $[\mathbf{X}_\mathrm{e}]_{n,k} = x_e[n-k]$, be a matrix composed of the even-indexed data used in the prediction. The vector of prediction errors is then given by

$$\mathbf{e} \;=\; \mathbf{x}_\mathrm{o} - \mathbf{X}_\mathrm{e}\mathbf{p}. \tag{9}$$

Our objective is to find the prediction coefficients that minimize the sum of squared prediction errors $\mathbf{e}^T\mathbf{e}$ while satisfying the $M < N$ polynomial constraints. Thus, we solve

$$\min_{\mathbf{p}} \|\mathbf{x}_\mathrm{o} - \mathbf{X}_\mathrm{e}\mathbf{p}\|^2 \quad \text{subject to (6)}, \tag{10}$$

with $\mathbf{V}^\diamond$ an $M \times M$ matrix determined as in Section 2.1. The optimal prediction coefficients for this constrained least squares problem can be efficiently computed using the QR factorization method [8, p. 567].

The optimal predictor effectively "locks-on" to the dominant signal structure at each scale. The wavelet coefficients $d[n]$ then represent the variations of the signal from this structure. Once the optimal predictor is determined, the update is designed using the methods of Section 2.2 to ensure that the dominant coarse-scale (low-frequency) structure is preserved in the coarse signal approximation that is used at the next scale.

3.2. Space-Adaptive Transforms

In addition to a scale-by-scale optimization, lifting permits us to *instantaneously* adapt the predictor to the signal and change the wavelet basis functions at each point and scale. In a space-adapted transform (SpAT), we employ the update/predict framework of Section 2.3 and choose a predictor from a suite of predictors to minimize each $d[n]$ value.

Our adaptive algorithm performs a $\widetilde{N} = 1$ point update, and then for each n chooses the $N \in \{1, 3, 5, 7\}$ point prediction that minimizes the value $d[n]$. We choose this $(N, \widetilde{N})$ pair, because the underlying wavelet functions are relatively smooth and the synthesis functions have small side-bands [9]. A demonstration of the SpAT applied to a step edge is shown in Figure 4. The transform is able to lock-on to the dominant signal structure at each point, and avoid discontinuities and other high-order polynomial phenomena that would decrease the quality of prediction.

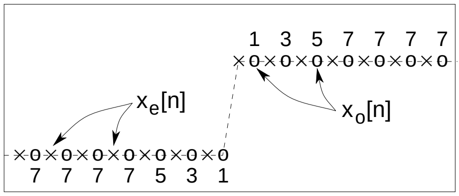

Figure 4: *In the SpAT, the order N of the predictor varies with space n to minimize the wavelet coefficient value $d[n]$. Above each $x[n]$ we give the corresponding choice $N(n)$. As the predictor approaches an edge, it decreases N (chooses wavelets of smaller spatial support) in order to avoid the edge.*

4. NUMERICAL EXPERIMENTS

In this section, we compare the performance of the new adaptive transforms with some of the standard Daubechies wavelets. Two experiments are performed. First, we compare the entropies of the coefficient distributions of several well-known test signals to assess the level of compaction afforded by the new transforms. Second, we compare the performance of the new transforms in a signal denoising application.

4.1. Entropy Comparison

The entropy of the transform coefficient distribution is a common measure of the efficiency of a signal transform [5]. If we collectively denote the scaling and wavelet coefficients by $\{w_i\}$, then the entropy is defined as

$$H(\mathbf{w}) \;=\; \sum_i |w_i|^2 \log_2 |w_i|^2, \qquad (11)$$

assuming the normalization $\sum_i |w_i|^2 = 1$.

Table 1 compares the entropies of the Daubechies-8 (D8) and Daubechies-2 (Haar) DWTs to those obtained using the ScAT and SpAT. The ScAT used an $N = \widetilde{N} = 4$ lifting construction, with $M = 3$ vanishing moments enforced. The first four signals, Doppler, Blocks, Bumps, and HeaviSine, are standard test signals introduced in [1]. The last signal, DoppelBlock, is a concatenation of the Doppler signal and the Blocks signal (hence it contains both smooth and edgy signal elements). All signal were 1024 samples long. The entropies in Table 1 show that both adaptive transforms perform nearly as well (or better) than the better of the D8 or the Haar in each test case.

Table 1: *Entropy results for various signals and transforms.*

Signal	**Entropy by Algorithm**			
	D8	*Haar*	*ScAT*	*SpAT*
Doppler	2.837	3.153	2.878	2.568
Blocks	2.598	2.618	2.517	2.318
Bumps	3.541	3.652	3.532	3.233
HeaviSine	2.262	2.500	2.250	2.265
DoppelBlock	3.543	3.597	3.414	3.149

4.2. Denoising Comparison

Because DWTs provide such a parsimonious representation for wide classes of signals, the DWT has proved to be a powerful tool for noise removal. The basic "wavelet denoising" programme [1] is described as follows. We observe L samples $\{x[n]\}$ of an unknown function f with additive i.i.d. Gaussian noises $\{\eta[n]\}$:

$$x[n] \;=\; s[n] + \eta[n], \qquad n = 0, 1, \ldots, L-1. \qquad (12)$$

We compute the DWT of x and apply a "soft-threshold" nonlinearity to the wavelet coefficients. The soft-threshold sets very small coefficients to zero and reduces all other coefficients by a fixed amount proportional to the standard deviation of the noise. The inverse DWT of the thresholded coefficients produces a "denoised" signal. For a more information see [1].

Table 2 provides the mean-squared error (MSE) performance of the four transforms and five signals discussed in Section 4.1 above. In this experiment, noise of standard deviation $0.1 \times \max_n |s[n]|$ was added to each of the test signals. The MSEs in Table 2 show again that both adaptive transforms perform nearly as well (or better) than the better of the D8 or the Haar in each test case.

5. CONCLUSIONS

This paper has described two new adaptive DWTs based on the lifting scheme. We used the lifting construction to adaptively match the DWT to a given signal based on data-based error criteria. Comparisons in entropy measures and signal denoising demonstrate the potential utility of the new transforms.

Table 2: *Denoising: MSE for various signals and transforms.*

Signal	**MSE by Algorithm**			
	D8	*Haar*	*ScAT*	*SpAT*
Doppler	0.0483	0.0733	0.0459	0.0521
Blocks	0.4428	0.3598	0.4084	0.3661
Bumps	0.4014	0.4403	0.3949	0.4102
HeaviSine	0.1966	0.4404	0.1990	0.2107
DoppelBlock	0.1111	0.1065	0.1110	0.0966

Many variations on the ideas presented here can be made to develop new adaptive DWTs. For example, a logical next step would be to combine the ScAT and SpAT into a scale *and* space adaptive transform. We also note that the new transforms are easily extended to images (a similar lifting construction exists for higher dimensional data). Finally, in this paper we have only examined the potential of the new transforms for signal denoising, but they may also improve algorithm performance in other applications, such as signal compression [9], detection, and classification.

Acknowledgements: Thanks to Geoff Davis and Wim Sweldens for introducing us to the lifting paradigm. Thanks also to Geoff for suggesting the update/predict architecture of Section 2.3.

6. REFERENCES

[1] D. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inform. Theory*, vol. 41, pp. 613–627, May 1995.

[2] I. Daubechies, *Ten Lectures on Wavelets*. New York: SIAM, 1992.

[3] W. Sweldens, "The lifting scheme: A custom-design contruction of biorthogonal wavelets," *J. Appl. Comp. Harm. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.

[4] A. H. Tewfik, D. Sinha, and P. Jorgensen, "On the optimal choice of a wavelet for signal representation," *IEEE Trans. Inform. Theory*, vol. IT–38, pp. 747–765, Mar. 1992.

[5] R. Coifman and V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inform. Theory*, vol. IT–38, pp. 713–718, Mar. 1992.

[6] F. J. Hampton and J. C. Pesquet, "A nonlinear subband decomposition with perfect reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing — ICASSP '96*, vol. 3, pp. 1523–1526, 1996.

[7] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," tech. rep., Bell Laboratories, 1996.

[8] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore: Johns Hopkins University Press, 1989.

[9] R. L. Claypoole, G. Davis, W. Sweldens, and R. G. Baraniuk, "Nonlinear wavelet transforms for image coding," in *Proc. 31st Asilomar Conf.*, (Pacific Grove, CA), 1997.