Flexible Wavelet Transforms Using Lifting

Roger L. Claypoole, Jr. and Richard G. Baraniuk, Rice University *

Summary

We introduce and discuss biorthogonal wavelet transforms using the lifting construction. The lifting construction exploits a spatial–domain, prediction–error interpretation of the wavelet transform and provides a powerful framework for designing customized transforms. We discuss the application of lifting to adaptive and non–linear transforms, transforms of non–uniformly sampled data, and related issues.

Introduction

Many applications (compression, analysis, denoising, etc.) benefit from signal representation with as few coefficients as possible. We also wish to characterize a signal as a series of course approximations, with sets of finer and finer details. The Discrete Wavelet Transform (DWT) provides such a representation.

The 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 [12, pp. 302–332]. The inverse DWT employs an inverse filterbank with lowpass filter $\widetilde{h}$ and highpass filter $\widetilde{g}$, as shown in Figure 1 For special choices of h, g, $\widetilde{h}$, and $\widetilde{g}$, the underlying wavelet and scaling functions form a biorthogonal wavelet basis [2].

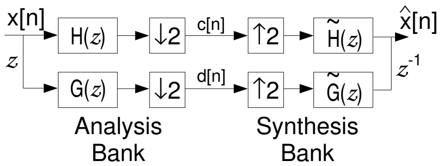

Fig. 1: *Filter Bank Representation of Wavelet Transform.*

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 follows 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, 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 enlightening 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 construct DWTs for non–traditional problems. The paper is organized as follows. First, we review the basic lifting construction, and generalize this construction to a non–uniform grid. Next, we describe applications of lifting to adaptive and non–linear wavelet transforms, along with a scheme for multi–scale interpolation to a uniform grid. We close with concluding remarks and plans for future work.

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 2):

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 $\mathcal{P}$:

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

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 $x_e[n]$:

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

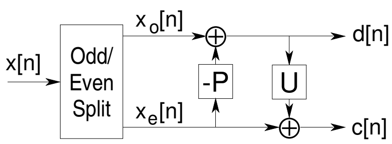

Fig. 2: *Lifting stage: Split, Predict, Update.*

*Supported by NSF, grant no. MIP–9457438, and Rice Consortium for Computational Seismic Interpretation. Email: clayporl@rice.edu, richb@rice.edu, Web: www.dsp.rice.edu

These three steps form a *lifting stage*. As seen in Figure 2, a lifting stage has the same structure as a filter bank ladder structure [5]. 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 nonlinear or space-varying, or the data is non–uniformly sampled.* Rearranging (1) and (2), we have

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

Predictor Design: We initially assume the data is uniformly sampled, and the prediction operator $\mathcal{P}$ is a linear shift-invariant filter, with z transform $P(z)$. In Figure 3, 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

$$\mathbf{g} \ = \ [-p_1,\ 0,\ -p_2,\ 1,\ -p_3,\ 0,\ -p_4]^T. \qquad (4)$$

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

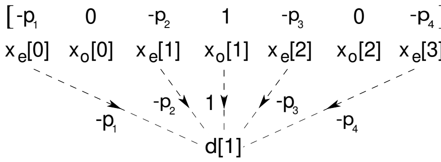

Fig. 3: *Prediction filtering. An $N = 4$ point linear prediction filter $P(z)$ yields the prediction vector $\mathbf{g}$ shown across the top.*

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 order $N-1$, we require that

$$\mathbf{V}\mathbf{g} \ = \ \mathbf{0}. \qquad (5)$$

This set of linear equations is readily solved, since $\mathbf{V}$ forms the first N rows of a Vandermonde matrix, which is always invertible [10]. 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 $\mathbf{p}$, the vector of coefficients of the prediction filter $P(z)$

$$\mathbf{V}^\diamond \mathbf{p} \ = \ [1, 0, \ldots, 0]^T. \qquad (6)$$

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 4, 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; $U(z) = u_1 z^{-2} + u_2 z^{-1} + u_3 + u_4 z$. In vector form, we have the equivalent filter $\mathbf{h}$ at the top of the Figure. Note that $\mathbf{h}$ is a function of both the update coefficients u_k and the prediction coefficients p_k.

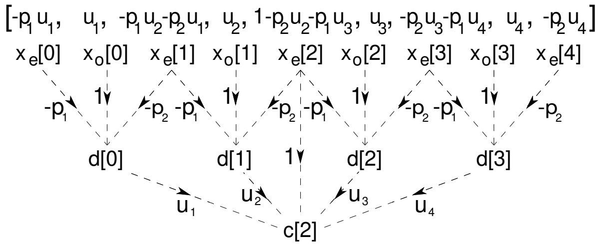

Fig. 4: *Update filtering. An $N = 2$ point linear predict followed by an $\widetilde{N} = 4$ point linear update yields the update vector* $\mathbf{h}$ *shown across the top.*

The update filter vector $\mathbf{h}$ should pass low-order polynomials into $c[n]$ while attenuating high-order polynomials. In the predictor design described above, we eliminated low order polynomials by requiring that $\mathbf{V}\mathbf{g} = \mathbf{0}$. The rows of $\mathbf{V}\mathbf{g}$ were monomial signals: $[\mathbf{V}]_{m,n} = n^m$, $n = -(N-1), \ldots, (N-1)$, $m = 0, 1, \ldots, N-1$. We design the update coefficients in similar fashion; as the prediction filter $\mathbf{g}$ eliminated low–order polynomials, the update filter $\mathbf{h}$ eliminates high–order polynomials. We create these high–order polynomials by taking low order monomials and superimposing alternating signs. Thus, we form the $\tilde{N} \times (2\tilde{N} - 1)$ matrix $\mathbf{V}'$ whose elements are $[\mathbf{V}']_{m,n} = (-1)^n n^m$, $n = -(N-1), \ldots, (N-1)$, $m = 0, 1, \ldots, N-1$. Since our goal is to attenuate these high order polynomials, we again require that

$$\mathbf{V}'\mathbf{h} \ = \ \mathbf{0}. \qquad (7)$$

This set of linear equations is also easily solved, since $\mathbf{V}'$ factors into the first N rows of a Vandermonde matrix and a diagonal matrix with all entries ± 1's. We rearrange 7 to get

$$\mathbf{V}^{\diamond\prime}\mathbf{u} \ = \ [1, 0, \ldots, 0]^T. \qquad (8)$$

For the example in Figure 4, we have

$$\mathbf{h} \ = \ [-p_1 u_1,\ u_1,\ (-p_1 u_2 - p_2 u_1),\ u_2,\ (1 - p_2 u_2 - p_1 u_3),$$
$$u_3,\ (-p_2 u_3 - p_1 u_4),\ u_4,\ -p_2 u_4]^T. \qquad (9)$$

Since the $N = 2$ prediction coefficients are already determined, there are $\widetilde{N} = 4$ unknowns (the update coefficients u_k) in $\mathbf{h}$. Solution of 8 yields these 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

[1] In fact, *all* wavelet transforms can be factored into a series of lifting stages (with perhaps multiple predicts and updates per stage) [9].

next coarser scale.[2] Since lifting is developed entirely in the space-domain, the lifted transform can be adapted to the signal borders instead of simple periodization or zero–padding [4]. Also, it is possible to eliminate and preserve fewer polynomials than we have prediction and update coefficients, respectively. This permits the incorporation of other problem–dependent constraints into the lifted wavelet transform. For example, the prediction operator could be designed to eliminate constants while minimizing the total variance of the detail coefficients [7].

Lifting Constraints and Vanishing Moments: The standard wavelet transform can be constructed as shown in Figure 1. By factoring the synthesis and analysis filters into their polyphase components [12, pp. 120–134], the transform can be constructed as shown in the the top of Figure 5. Also, by combining the prediction and update steps, the lifted wavelet transform can be implemented as shown in the bottom of Figure 5 (assuming linear operators and a uniform sampling grid).

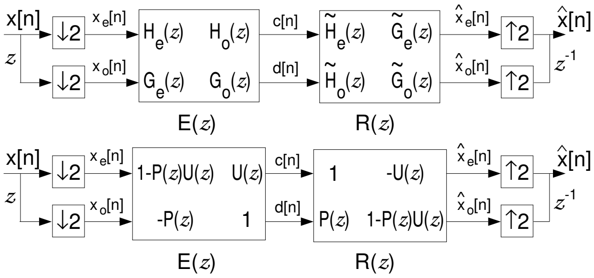

Fig. 5: *Wavelet Transform: Polyphase Form (top) and Lifting–Combined Steps (bottom).*

We equate the entries in these polyphase matrices, yielding the following relations:

$$H(z) = H_e(z^2)+z^{-1}H_o(z^2) = 1-P(z^2)U(z^2)+z^{-1}U(z^2)$$
$$G(z) = G_e(z^2)+z^{-1}G_o(z^2) = -P(z^2) + z^{-1}$$

with similar expressions for $\widetilde{H}(z)$ and $\widetilde{G}(z)$.

We have the wavelet filters in terms of the prediction and update operator coefficients. Since this is a wavelet system, these filters must satisfy a recurrence relation [2], and as we add vanishing moments to the underlying wavelet functions, we place constraints on the lifting operators. For example, if we add a zeroth vanishing moment to $\psi(t)$ we have:

$$\int \psi(t)\,dt = \int \sum_k g[k]\phi(2t-k)\,dt = 0 \qquad (10)$$

To satisfy this equation, we must have $\sum_k g[k] = 0$, which, in turn, forces $p_1 + p_2 \cdots + p_N = 1$. This is the first prediction constraint (the prediction must eliminate zeroth order polynomials)! Every vanishing moment we add to $\psi(t)$ (and, by the biorthogonal structure, to $\widetilde{\phi}(t)$) is equivalent to eliminating higher order polynomials in our prediction step. In similar fashion, adding vanishing moments to $\widetilde{\psi}(t)$ and $\phi(t)$ is equivalent to the linear update constraints.

[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.

Thus, *adding vanishing moments is equivalent to eliminating and preserving polynomials with the predict and update steps, respectively.* Both interpretations yield identical constraints on the wavelet filters h, g, $\widetilde{h}$ and $\widetilde{g}$. However, the lifting scheme never explicitly utilizes the polyphase representation or the underlying scaling and wavelet functions, and therefore makes the incorporation of non–linearities and adaptivity into the wavelet transform more understandable. If lifting constraints are traded for other problem–dependent constraints, it is clear that vanishing moments in the underlying wavelet functions are being surrendered.

Irregular Sampling

Since the lifted wavelet transform is a spatially–based construction, the extension to non–uniformly sampled data is straightforward. We present here an alternate interpretation to that developed by Sweldens [4]. We have data $x[n]$, but the data are indexed by $i[n]$. That is, $x[0]$ lies at coordinate $i[0]$, $x[1]$ lies at coordinate $i[1]$, etc. Although any split of this data is possible[3], we again choose an odd/even split, where $x_e[n] = x[2n]$, $x_o[n] = x[2n+1]$.

Predictor Design: As stated earlier, the goal of the prediction step is to eliminate all low-order polynomials from $x[n]$ in creating the wavelet coefficients. This was accomplished by creating a set of prediction coefficients p_k which predict each $x_o[n]$ as a low–order polynomial version of its even neighbors.

For irregularly sampled data, the same mechanics apply. Instead of uniform samples of the low order polynomial, we are given samples at irregular indices $i_e[n]$ (the indices of the even data). We trace the prediction coefficients through the same tree shown in figure 3. Then, we find the coefficients by requiring the prediction vector $\mathbf{g}$ to eliminate low order polynomials. However, in this case, the low order polynomials are sampled at irregular intervals. Thus, we again form the $N \times (2N-1)$ matrix $\mathbf{V}$, but its rows are now monomial signals sampled at the appropriate indices: $[\mathbf{V}]_{m,n} = i[n]^m$, $n = -(N-1), \ldots, (N-1)$, $m = 0, 1, \ldots, N-1$. (Again, $0^0 = 1$.) We require that

$$\mathbf{V}\mathbf{g} \;=\; \mathbf{0}. \qquad (11)$$

so that the predictor suppresses all polynomials up to order $N-1$. Again, the solution exists in an N-dimensional subspace, so we have

$$\mathbf{V}^{\diamond}\mathbf{p} \;=\; [i_o[k]^0, i_o[k]^1, \ldots, i_o[k]^{N-1}]^T. \qquad (12)$$

$\mathbf{V}^{\diamond}$ is always invertible; the only change from equation 5 is the right hand side, which is now a function of the index $i_o[k]$.

[3] Including optimizing the split to minimize average distance, or blocking the data and splitting by vectors [11].

Update Design: This update design is identical to that described earlier, except we have irregular coordinates. We again trace the update coefficients up through the tree in figure 4, and create an update vector $\mathbf{h}$ with elements at the coordinates $i[n]$. We desire to attenuate high-order polynomials, so we apply $\mathbf{h}$ to monomials with alternating signs (these monomials are evaluated at the $i[n]$). We collect terms and solve for the $\widetilde{N}$ update coefficients u_k.

Extensions

Adaptive and Non-linear Transforms: Lifting is a "natural" interpretation of the wavelet transform, entirely in the spatial domain. Thus, lifting easily allows the incorporation of adaptivity into the prediction step. The lifting construction guarantees perfect reconstruction regardless of the choice of prediction and update operators. However, when we incorporate adaptive predictors, we need to make a modification to retain the coarse characteristics of the scaling coefficients.

The Update/Predict Programme: In the lifting framework of Figure 2, the update structure depends on the predictor structure. Hence, if $\mathcal{P}$ is space-varying or nonlinear, then so is $\mathcal{U}$, and the update design procedure becomes unwieldy. A crafty detour around this problem is to perform the update step first, followed by the prediction [6]. The relevant equations then become

$$c[n] = x_e[n] + \mathcal{U}(x_o[n]), \qquad d[n] = c[n] - \mathcal{P}(x_e[n]). \quad (13)$$

After designing a linear update filter to preserve the first $\widetilde{N}$ low-order polynomials in the data, we can apply any 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. Also, the scaling coefficients $c[n]$ can be quantized and reconstructed before the $d[n]$ are computed [6]. This permits a signal–adaptive predictor without propagation of quantization error.

Multiscale Interpolation to Uniform Grid: Lifting can be used to construct a multiscale analysis for a non–uniformly sampled signal. Since the lifting construction is based on polynomial interpolation [4], this suggests a lifting algorithm to interpolate data to a uniform grid [13].

In the prediction step, the prediction filter coefficients p_k are chosen to fit a low order polynomial over a window of the even data $x_e[n]$. We interpolate this polynomial at an odd coefficients $x_o[n]$, and the detail coefficient is the failure of $x_o[n]$ to be predicted by the local polynomial. However, we could interpolate this low order polynomial to any point; not just at $x_o[n]$. Thus, the detail coefficients could be constructed on a regular grid, with each coefficient the failure of *an interpolation of* $x_o[n]$ to be predicted by its nearest even neighbors. These detail coefficients could then be used to update the $x_e[n]$ to create the scaling coefficients $c[n]$; these coefficients need not be interpolated to the uniform grid.

The next iteration of the wavelet transform would create the next set of detail coefficients *on the uniform grid*, and continue in this fashion until all the data have been interpolated. We speculate that application of the appropriate inverse lifted transform (on the uniform grid) should yield a uniformly sampled version of our non–uniformly sampled data.

Conclusions

For many applications, the traditional wavelet transform is not well suited. Lifting increases our flexibility while retaining the useful properties of the traditional wavelet transform. Also, lifting is an entirely space–domain construction. Thus, we can use lifting to accommodate *non–traditional* problems, including non–uniformly sampled data. In this paper we presented such an algorithm, and discussed extensions to adaptive and non–linear transforms. We also proposed an algorithm for multi–scale interpolation of non–uniformly sampled data to a uniform grid.

Acknowledgments: Thanks to Geoff Davis and Wim Sweldens for introducing us to the lifting paradigm. Thanks also to Geoff for suggesting the update/predict architecture. Thanks to Rob Nowak for assistance with non–linear transforms.

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] W. Sweldens and Peter Schröder, "Building Your Own Wavelets at Home," tech rep., Industrial Mathematics Initiative, Mathematics Department, University of South Carolina, no. 1995:5, 1995.

[5] 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.

[6] 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.

[7] R. L. Claypoole, R. G. Baraniuk, and R. D. Nowak, "Adaptive Wavelet Transforms via Lifting," to appear in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing — ICASSP '98.*

[8] V. J. Mathews, "Adaptive polynomial filters," *IEEE SP Magazine*, vol. 8, no. 3, pp. 10-26, 1991.

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

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

[11] X. G. Xia and B. W. Suter, "Multirate Filter Banks with Block Sampling," *IEEE Trans. Signal Processing*, vol. 44, no. 3, pp. 484–496, March 1996.

[12] A. K. Soman and P. P. Vaidyanathan, “On Orthonormal Wavelets and Paraunitary Filter Banks”, *IEEE Transaction on Signal Processing*, Vol. 41, No. 3, pp 1170–1183, March 1993.

[13] Kyeokho Choi, "Technical Paper", Rice University, 1998.