

# Nonlinear Wavelet Transforms for Image Coding via Lifting

Roger L. Claypoole, Jr., *Senior Member, IEEE*, Geoffrey M. Davis, Wim Sweldens, *Member, IEEE*, and  
Richard G. Baraniuk, *Fellow, IEEE*

**Abstract**—We investigate central issues such as invertibility, stability, synchronization, and frequency characteristics for nonlinear wavelet transforms built using the lifting framework. The nonlinearity comes from adaptively choosing between a class of linear predictors within the lifting framework. We also describe how earlier families of nonlinear filter banks can be extended through the use of prediction functions operating on a causal neighborhood of pixels. Preliminary compression results for model and real-world images demonstrate the promise of our techniques.

**Index Terms**—Adaptive signal processing, image coding, wavelet transforms.

## I. INTRODUCTION

IN HIS CLASSIC treatise on the workings of the human visual system, Marr focused on the importance of the *representation* of information for various cognitive tasks [1]. The way in which information is represented brings out certain types of features while hiding others. Image compression applications also rely heavily on having an efficient representation of image data. Ideally we would like to approximate an image with a small number of parameters; the wavelet transform provides such an efficient representation [2].

Transform coding consists of three components: a reversible, linear transform to map the image into a set of transform coefficients; nonreversible quantizers; and an encoder [3], [4]. Typically, a significant number of the transform coefficients are small, and can therefore be coarsely quantized or completely discarded, with little distortion. Compression is achieved during the quantization and encoding of the transformed coefficients, and not during the transformation step.

In this paper we focus on improving the properties of the transform rather than the encoder, expanding on our work in [5].

Manuscript received September 14, 1999; revised March 18, 2002. This work was supported by the NSF under Grant MIP-9457438, by the ONR under Grant N00014-99-1-0813, by DARPA/AFOSR under Grant F49620-97-1-0513, and by the Texas Instruments Leadership University Program. The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Touradj Ebrahimi.

R. L. Claypoole is with the Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH 45433-7765 USA (e-mail: r.claypoole@ieee.org).

G. M. Davis is with Sigma Xi, The Scientific Research Society, Research Triangle Park, NC 27709 USA (e-mail: gdavis@sigmaxi.org).

W. Sweldens is with Lucent Technologies, Bell Laboratories, Murray Hill, NJ 07974 USA (e-mail: wim@bell-labs.com).

R. G. Baraniuk is with the Department of Electrical and Computer Engineering, Rice University, Houston, TX 77005-1892 USA (e-mail: richb@rice.edu).

Digital Object Identifier 10.1109/TIP.2003.817237

More precisely, we will construct adaptive wavelet transforms that result in fewer large wavelet coefficients. Such nonlinear wavelet transforms provide added flexibility for image representations.

Until recently, the wavelet transforms used for image compression were constructed with linear filter banks. Construction of nonlinear filter banks was proposed in [6], [7]. The experiments with a nonlinear filter bank for image coding presented in [8] are promising. The key open question in the use of these nonlinear constructions is one of design: what is the most effective way to utilize the additional degrees of freedom obtained from relaxing the constraint of linearity?

We examine issues such as invertibility, stability, artifacts, and frequency-domain characteristics (to the extent to which these are well-defined) in the construction of nonlinear wavelet transforms. Our analysis builds on the new perspective provided by the lifting framework [9], [10] for the wavelet transform. The lifting framework allows us to incorporate nonlinearities while retaining control over the properties of the wavelet transform. The nonlinearity comes from adaptively choosing from a set of linear predictors. We also show how the family of nonlinear filter banks of [6], [7] can be extended through the use of prediction functions operating on a causal neighborhood.

Our paper is organized as follows. In Section II, we review the wavelet transform and the lifting construction, and show how to introduce adaptivity into the transform. In Section III, we discuss issues surrounding adaptivity, and in Section IV we propose an edge-avoiding adaptive transform. In Section V, we demonstrate this transform via compression of artificial and real-world images. We conclude in Section VI and propose ideas for future research.

## II. WAVELETS AND THE LIFTING SCHEME

### A. Wavelets

The discrete wavelet transform represents a signal in terms of shifts and dilations of a low-pass scaling function  $\phi(t)$  and a bandpass wavelet function  $\psi(t)$  [2]. The transform is multi-scale, in that it creates a set of coarse coefficients that represent signal information at the lowest scale, and sets of detail coefficients with increasingly finer resolution. The transform is typically implemented as a filter bank with analysis low-pass filter  $H(z)$  and high-pass filter  $G(z)$ , as shown in Fig. 1. The inverse transform uses synthesis low-pass  $\tilde{H}(z)$  and high-pass  $\tilde{G}(z)$ , as shown in Fig. 2. For special choices of  $H$ ,  $G$ ,  $\tilde{H}$ , and  $\tilde{G}$ , the underlying wavelets and scaling functions form a biorthogonal basis and provide perfect reconstruction [2]. The transform is

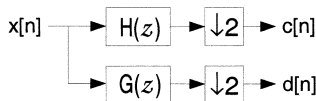


Fig. 1. Filter bank implementation of the wavelet transform.  $H$  and  $G$  are the analysis low-pass/high-pass pair.  $c[n]$  and  $d[n]$  are the scaling and wavelet coefficients, respectively.

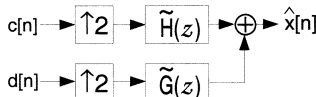


Fig. 2. Filter bank implementation of the inverse wavelet transform. With appropriate choices of  $\tilde{H}$  and  $\tilde{G}$ , the transform will yield a perfectly reconstructed output sequence.

typically iterated on the output of the low-pass band ( $c[n]$ ) to create the series of detail coefficients at different scales.

The wavelet representation is efficient because images are often well modeled as a set of locally smooth regions separated by edges. Within these smooth regions, fine-scale wavelet coefficients are small, and coefficients decay rapidly from coarse to fine scales. In the neighborhood of edges wavelet coefficients decay much more slowly, but because of the local support relatively few wavelet coefficients are affected by edges. However, these large wavelet coefficients near edges are expensive to code. Many image coders are designed to operate on wavelet transformed data, and much current research focuses on enabling these coders to exploit the structure present in wavelet coefficients along edges. Current successful coders perform some form of conditioning [11], variance prediction [12], or context-based entropy coding [13].

### B. Lifting Scheme

Lifting [9], [10] was originally developed to adjust wavelet transforms to complex geometries and irregular sampling leading to so-called *second generation wavelets*. It can also be seen as an alternate implementation of classical, first generation wavelet transforms [9], [14]. The main feature of lifting is that it provides an entirely spatial-domain interpretation of the transform, as opposed to the more traditional frequency-domain based constructions. The local spatial interpretation enables us to adapt the transform not only to the underlying geometry but also to the data, thereby introducing nonlinearities while retaining control of the transform's multi-scale properties.

A typical lifting stage is comprised of three steps: Split, Predict, and Update (as shown in Fig. 3):<sup>1</sup>

*Split:* Let  $x[n]$  be a signal. We first *split*  $x[n]$  into its even and odd polyphase components  $x_e[n]$  and  $x_o[n]$ , where  $x_e[n] = x[2n]$  and  $x_o[n] = x[2n + 1]$ . In this paper we work only with the even and odd polyphase components of  $x[n]$ , but in principle any partition of  $x[n]$  into nonoverlapping sets is possible [10]. If the  $x[n]$  correspond to the samples of an underlying smooth, slowly varying function, then the even and odd polyphase components are highly correlated. This correlation structure is typ-

<sup>1</sup>In Fig. 3, the outputs of the lifting stage are weighted by  $k_e$  and  $k_o$ . These values serve to normalize the energy of the underlying scaling and wavelet functions, respectively. Thus, this normalization could be considered a fourth lifting step.

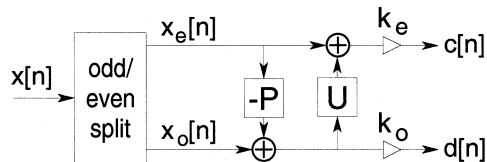


Fig. 3. Typical lifting steps: Split, Predict, and Update.

ically local, and thus we should be able to accurately predict each odd polyphase coefficient from the nearby even polyphase coefficients.

*Predict:* In the interpolating formulation of lifting, we *predict* the odd polyphase coefficients  $x_o[n]$  from the neighboring even coefficients  $x_e[n]$ . The predictor for each  $x_o[n]$  is a linear combination of neighboring even coefficients

$$P(x_e)[n] = \sum_l p_l x_e[n + l]. \quad (1)$$

We obtain a new representation of the  $x[n]$  by replacing  $x_o[n]$  with the prediction residual. This leads to the first lifting step

$$d[n] = x_o[n] - P(x_e)[n]. \quad (2)$$

If the underlying signal is locally smooth, the prediction residuals  $d[n]$  will be small. Furthermore, the new representation contains the same information as the original signal  $x[n]$ : given the even polyphase  $x_e[n]$  and the prediction residuals  $d[n]$ , we can recover the odd polyphase coefficients  $x_o[n]$  by noting that

$$x_o[n] = d[n] + P(x_e)[n]. \quad (3)$$

This prediction procedure is equivalent to applying a high-pass filter to  $x[n]$ . The prediction filter is typically designed to exactly predict local polynomials up to and including degree  $N - 1$ . In wavelet terminology, the underlying synthesis scaling function corresponding to this prediction filter can reproduce polynomials of degree up to  $N - 1$ , and the dual (analysis) wavelet has  $N$  zero moments.

*Update:* The third lifting step transforms the even polyphase coefficients  $x_e[n]$  into a low-pass filtered and subsampled version of  $x[n]$ . We obtain this coarse approximation by *updating*  $x_e[n]$  with a linear combination of the prediction residuals  $d[n]$ . We replace  $x_e[n]$  with

$$c[n] = x_e[n] + U(d)[n], \quad (4)$$

where  $U(d)$  is a linear combination of neighboring  $d$  values

$$U(d)[n] = \sum_l u_l d[n + l]. \quad (5)$$

Each lifting step is always invertible; no information is lost. Assuming the same  $P$  and  $U$  are chosen for the analysis and synthesis stages, the lifting construction guarantees perfect reconstruction for any  $P$  and  $U$ . Given  $d[n]$  and  $c[n]$ , we have

$$x_e[n] = c[n] - U(d)[n] \quad (6)$$

and  $x_o[n]$  from (3).

The inverse lifting stage is shown in Fig. 4. Note that  $c$  and  $d$  are at half rate, and thus this transform corresponds to a criti-

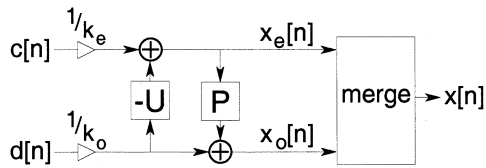


Fig. 4. Typical inverse lifting steps: undo the update, undo the predict, and merge.

cally sampled perfect reconstruction filter bank. One can show that the update function determines the properties of the dual wavelet and primal scaling function. In particular, if the update filter is one-half the adjoint of the predict filter, then the primal (synthesis) wavelet has  $N$  zero moments as well [9].

### C. Examples

A simple example of lifting is the construction of the Deslauriers-Dubuc family of wavelets [9] from a single Deslauriers-Dubuc [15] prediction step followed by a single update step. For example, the following prediction and update steps comprise a single stage of the (4,4) Deslauriers-Dubuc wavelet transform

$$d[n] = x_o[n] - \frac{(-x_e[n-1] + 9x_e[n] + 9x_e[n+1] - x_e[n+2])}{16}, \quad (7)$$

$$c[n] = x_e[n] + \frac{(-d[n-2] + 9d[n-1] + 9d[n] - d[n+1])}{32}. \quad (8)$$

The predict step cancels cubic polynomials and leaves the residual in the high-pass signal  $d[n]$ . The update step results in a low-pass and subsampled version of  $x[n]$  being placed in  $c[n]$ . It should be emphasized that lifting is a general construction and not limited to the Deslauriers-Dubuc family. Using the Euclidean algorithm, we can decompose any FIR wavelet transform into a sequence of prediction and update steps [14]. Thus, the lifting implementation shown in Fig. 3, with possibly multiple stages, is equivalent to the filter bank implementation of the wavelet transform shown in Fig. 1. Unfortunately at this point we do not have a spatial interpretation for general transforms factored into lifting steps; consequently we currently do not know how to make adaptive versions of general wavelet transforms.

A second example of a nonlinear lifting construction is the integer-to-integer S+P transform of Said and Pearlman [16], shown in Fig. 5. The outputs  $h[n]$  and  $c[n]$  of the S algorithm are computed as

$$h[n] = x_o[n] - x_e[n] \quad (9)$$

$$c[n] = x_e[n] + Q\left(\frac{h[n]}{2}\right) \quad (10)$$

where  $Q$  is a round-off operator to ensure the transform is integer-to-integer. The P transform creates the detail coefficients  $d[n]$  as

$$d[n] = h[n] - Q(P(c[n]) + P_b(h[n])). \quad (11)$$

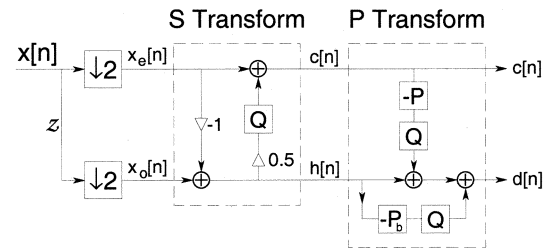


Fig. 5. Said and Pearlman (S+P) transform [16].

As shown at the bottom of Fig. 5, the S+P transform includes an additional prediction operator  $P_b$  which is outside the rungs of the “ladder.” This is an optional filter, and must be causal to ensure that the inverse transform can be implemented with identical filters to the forward transform.

In [17], it was shown that the S+P transform can be seen as a three-step nonlinear lifted transform. The S transform is constructed as a one point predict followed by a one point update. The P transform is an additional prediction step, combined with the causal  $P_b$  filter. The nonlinearity comes from the quantizers which are needed to ensure an integer-to-integer transform. Due to the nature of the lifting implementation (and the causality of  $P_b$ ), perfect reconstruction is guaranteed despite the presence of the nonlinear quantizers. It is interesting to note that the optimized coefficients of the P transform proposed in [16] satisfy some of the linear lifting constraints discussed in Section II-B.

### D. Introducing Adaptivity Into the Wavelet Transform

Wavelet bases typically employed for image compression (such as the Daubechies (9,7) system [2]) utilize smooth scaling and wavelet functions. Such bases can be easily constructed with the predict-then-update form of lifting described above. Larger predictors (predictors that can exactly predict polynomials of higher degree) correspond to smoother basis functions; these lifting predictors work well when the underlying signal is smooth (just as the Daubechies (9,7) system works best when the signal is smooth).

However, most images consist of regions of smoothness and texture separated by discontinuities (edges). These discontinuities cannot be well-represented by smooth basis functions. Since smooth basis functions correspond to lifting predictors with wide support, these predictors work poorly near edges, when the discontinuity is within the data we are using for the prediction.

Our goal is to introduce a mechanism that allows us to choose the prediction operator based on the local properties of the image. This makes the P operator data-dependent and thus nonlinear. However, lifting guarantees that the transform remains reversible. In regions where the image is locally smooth, we use higher order predictors. Near edges we reduce the order and thus the length of the predictor. This avoids making a prediction based on data which is separated from the point of interest by a discontinuity. Ideally we would like to use predictors that take into account the fact that discontinuities in images tend to occur along continuous curves. Such an adaptation would allow us to exploit the additional spatial structure that we know exists in edges.

### III. FILTER DESIGN

Adapting the predictor makes our transform nonlinear. However, the concept of basis functions relies fundamentally on linear superposition. Consequently, the notion of a single basis function no longer makes sense for nonlinear transforms. We thus focus on the spatial properties of the transform when designing our predictors.

#### A. Multi-Resolution Properties

When the prediction and update operators are constructed via the polynomial lifting constraints, the output of the update step is a coarse approximation (low-pass and downsampled) version of our image. We need this coherent interpretation of the update coefficients, since they will be input to further iterations of the transform. After the first iteration, all subsequent predictions are based on updated coefficients. If we are to make effective prediction throughout the transform, we need some kind of structure in the update. However, if the prediction is performed with a nonlinear operator, it may not be possible to construct an update operator that satisfies the polynomial lifting constraints and provides a low-pass interpretation of the updated coefficients.

Consider again the example (8). While it is easy to see that the prediction filter  $P$  leads to a high-pass filter, it is not immediately clear that the update  $U$  leads to a low-pass filter. The reason is that the lifting structure mandates that the high-pass coefficients  $d$  must be reused in the computation of  $c$ , and thus  $c$  depends both on  $P$  and on  $U$ . By carefully adjusting the update  $U$  to the prediction  $P$ , we can ensure that  $c$  is a low-pass-filtered and subsampled version of the original signal. In the example,  $U(d)$  had to be chosen as  $(d[n-2] + 9d[n-1] + 9d[n] + d[n+1])/32$ . While we know how to adjust  $U$  for a spatially varying, but linear  $P$  [10], it is not immediately clear how to construct a nonlinear  $U$  that preserves frequency localization (to the extent that this is well-defined) when we have a nonlinear  $P$ .

#### B. Stability and Synchronization

We also need to ensure that the transform is stable. Lossy coding schemes introduce errors into the transform coefficients, so it is crucial that the nonlinearities do not unduly amplify these errors. Our goal is to use a high-order predictor in smooth regions and a low-order predictor near edges. In order to avoid sending side information on which predictor was chosen, we need to base the choice only on the  $x_e[n]$ . However, in lossy compression the decoder only has the quantized even coefficients  $\hat{x}_e[n]$  rather than the original coefficients  $x_e[n]$ . If we use locally adapted filters, then quantization errors in coarse scales could cascade across scale and cause a series of incorrect filter choices leading to serious reconstruction errors.

In the predict-then-update case, the problem of stability cannot be solved by synchronization alone, i.e., having the encoder make its choice of predictor based on quantized data. The reason is that the reconstructed values  $\hat{x}_e[n]$  are obtained from quantized low-pass values  $\hat{c}[n]$ . The low-pass signal  $c[n]$  is a function of the prediction residual signal  $d[n]$ , which in turn depends on what filters are chosen for prediction, as shown on the left in Fig. 6. Hence the encoder cannot obtain the quantized values  $\hat{x}_e[n]$  until it selects a predictor, and it cannot

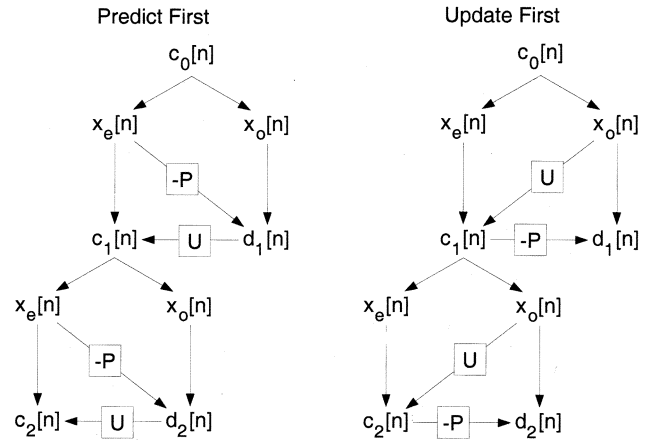


Fig. 6. Two-iteration lifted wavelet transform trees with predict-first (left) and update-first (right). When predicting first, the prediction must be performed prior to construction of the coarse coefficients and iteration to the next scale. When updating first, the prediction operator is outside the loop. The coarse coefficients can be iterated to the lowest scale, quantized, and reconstructed prior to the predictions.

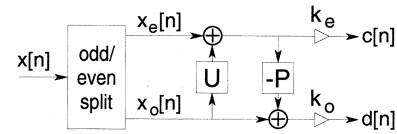


Fig. 7. Update-first lifting sequence.

select a predictor without obtaining  $\hat{x}_e[n]$ . If we are to employ a nonlinear lifting procedure for lossy coding, it is essential that we avoid this Catch 22.

#### C. Solution: Update First

We propose a simple modification that solves the stability and synchronization problems: reverse the order of the predict and update lifting steps in the wavelet transform (see Fig. 7). We first update the even samples based on the odd samples yielding the low-pass coefficients  $c[n]$ . We then reuse these low-pass coefficients to predict the odd samples, which gives the high-pass coefficients  $d[n]$ . We use a linear update filter and let only the choice of predictor depend on the data.

Because we update first and the transform is only iterated on the low pass coefficients  $c[n]$ , all  $c[n]$  throughout the entire pyramid linearly depend on the data and are not affected by the nonlinear predictor. This is shown in Fig. 6. The tree on the left shows the predict first pyramid. Clearly, it is impossible to create the coarse coefficients without first using the prediction operator to create the detail coefficients. However, in the update-first tree on the right, the prediction operators are not in the loop. Thus the prediction is only based on low-pass coefficients that are computed as in the classical wavelet transform. Furthermore, if we perform the transform *backward*, i.e., starting the prediction process at the lowest frequency (coarsest) subband and working from coarse to fine scales, we can keep the encoder and decoder perfectly synchronized. The predictor operates on raw data, but the *choice of predictor* is based on quantized data. This ensures that the encoder and decoder are choosing predictors based on the same data, and eliminates propagation of error due to incorrect decisions at the decoder. Moreover the low-pass branches of

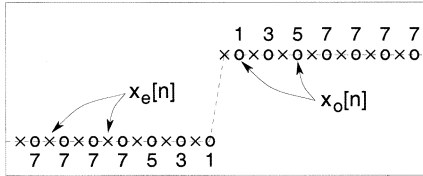


Fig. 8. Predictor selection at an ideal step edge. Numbers indicate the order of the predictors used. The closer to the edge, the lower the order of the predictor.

our entire multi-resolution scheme now are linear. Consequently we still have the notion of a dual (analysis) scaling function.

Our update-then-predict lifting scheme is related to the Laplacian pyramid of Burt and Adelson [18], in which images are represented as a series of prediction residuals, and the predictors are not constrained to being linear. The Laplacian pyramid has the disadvantage that it expands the number of coefficients in the image being transformed by a factor of  $4/3$ . Lifting, on the other hand, guarantees a critically sampled decomposition.

Our implementation is also similar to the framework developed independently by Gerek and Cetin in [19]. However, by constructing our transform via the lifting framework, we are able to incorporate adaptivity while retaining control over the underlying properties of the transform.

#### IV. ADAPTIVE WAVELET TRANSFORM

We now have a framework for introducing adaptivity into the wavelet transform. We will create and quantize all the coarse coefficients to the lowest scale (update first), and then adapt the prediction operator  $P$  to these coefficients. The question remains on how to determine the appropriate  $P$ .

##### A. Edge-Avoiding Prediction

As stated in Section II-D, our goal is to choose the prediction operator based on the local properties of the image. For each prediction window, we analyze the data to determine if it is well approximated by a low order polynomial. If it is, then we use a high-order predictor with wide support, which corresponds to a smooth basis function. If the data does not meet our smoothness criteria, we determine which pixels in the prediction window contribute to the failure. We classify these pixels as “edge” or discontinuity coefficients. Near these edges we reduce the order of the predictor so that the neighborhood we use for prediction never overlaps the edge. In this manner we maintain high accuracy away from edges, and avoid large errors in the presence of edges. Fig. 8 illustrates the process of selecting these predictors near an ideal step edge.

##### B. Choice of Prediction Filters

The question remains on how to find the  $P$  and  $U$  filters even in the linear case. One choice is the same  $P$  and  $U$  filters from the Deslauriers-Dubuc family, except use  $P$  for the update, followed by  $U$  for the prediction (with appropriate normalization). Swapping  $P$  and  $U$  in this fashion reverses the roles of the analysis and synthesis functions. However, this is problematic for coding applications, because the analysis wavelets in the Deslauriers-Dubuc family are much less smooth than the synthesis wavelets [2]. Since reconstructed images are built up

from synthesis wavelets, these nonsmooth building blocks lead to highly visible artifacts in the reconstructed image when the coefficients are quantized.

It is possible to boost the smoothness of the new building blocks by increasing the size of the filters (and adding more vanishing moments to the underlying scaling and wavelet functions). However, due to the biorthogonal structure of the update-first, single-stage lifting construction, the size of the synthesis filter  $\tilde{H}(z)$  will always be larger than that of the analysis filter  $H(z)$ . We observe that this leads to excessive ripple in the new building blocks, which in turn causes ringing in our reconstructed image.

Instead we propose a solution based on Donoho’s average-interpolation that fits into the update-predict form of lifting [20], [21]. This leads to the  $(1, N)$  branch of the Cohen-Daubechies-Feauveau family which is biorthogonal to the box function [22]. This family of wavelets can all be implemented with an update-first architecture. Let us consider a simple example. The low-pass coefficients are first computed using a Haar filter (a one-point update filter)

$$c[n] = \frac{(x[2n] + x[2n + 1])}{2}. \quad (12)$$

The high-pass coefficients are the residuals of a prediction of the odd samples based on the  $c[n]$ . The first-order Haar prediction (leading to the  $(1, 1)$  wavelet) is

$$d[n] = x[2n + 1] - c[n], \quad (13)$$

while the third-order predictor, i.e., one that is exact for quadratics and leads to the  $(1, 3)$  wavelet, is given by

$$d[n] = x[2n + 1] - \left( -\frac{c[n - 1]}{8} + c[n] + \frac{c[n + 1]}{8} \right). \quad (14)$$

Predictors of higher order can be built in a straightforward way. The smoothness of the resulting scaling functions increases with the order. A lower bound for the Hölder regularity  $R(N)$  as a function of  $N$  is given by  $R(3) = 0.678$ ,  $R(5) = 1.272$ ,  $R(7) = 1.826$ ,  $R(9) = 2.354$ , and asymptotically  $R(N) \approx 0.2075N$  [20]. The scaling and wavelet functions for the  $(1, 7)$  set are shown in Fig. 9. These functions correspond to an update-first architecture; in this case a one-point update followed by a seven-point prediction. In numerical experiments this filter set yields compression performance approaching that of the Daubechies  $(9, 7)$  filter set that is more commonly used in image coding applications.

This nonlinear lifting framework generalizes the ideas of de Quieroz *et al.* [8] and makes clear the relationship between the nonlinear filter banks described by these authors and the wavelet transform. The filter bank described in [8] generates the high-pass subbands using a nonseparable median filter, and the low-pass subbands via down-sampling. This filter bank performs particularly well for test images containing sharp edges, such as the *cameraman* image and text; it minimizes problems with ringing around the edges. However, the transform suffers from speckling artifacts due to aliasing of high frequency noise into the low-pass subbands. Our use of an anti-aliasing function via lifting has the potential to eliminate this speckling while maintaining high quality reconstruction around edges.

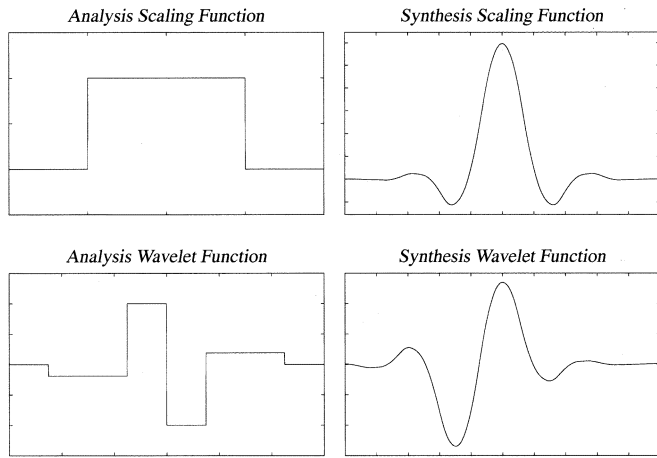


Fig. 9. Scaling and wavelet functions for the order (1,7) Cohen-Daubechies-Feauveau filter used in our experiments. These basis functions correspond to the update first form of lifting.

The idea of adaptively choosing from the  $(1, N)$  family of filters is similar to the work done independently by Boulgouris *et al.* [23], who use the  $(N, 2)$  family (predict first) to provide additional vanishing moments in the synthesis wavelet function for improved *lossless* image compression. However, our update first architecture provides for encoder/decoder synchronization (see Section IV-C, below) despite our application to *lossy* compression.

### C. Synchronization

As we stressed in Section III-C, maintaining synchronization between the adaptations of the encoder and the decoder is essential for a stable inversion. Encoding a  $j$ -level transform proceeds as follows: we first compute the coarsest scale coefficients of the transform  $c^j[n]$  by iterating the linear update procedure  $j$  times. We quantize  $c^j[n]$  to  $\hat{c}^j[n]$  and transmit them. Then we compute the high-pass coefficients  $d[n]$  as

$$d[n] = c^{j-1}[2n+1] - P_{\hat{c}^j}^j(c^j)[n], \quad (15)$$

quantize them to  $\hat{d}[n]$  and transmit them. Although the predictor operates on the unquantized  $c^j[n]$ , the *choice* of predictor,  $P_{\hat{c}^j}^j$ , is based on the quantized data  $\hat{c}^j[n]$ . Both encoder and decoder now need the quantized values of the next finer scale  $\hat{c}^{j-1}$ ; the even and odd components are respectively computed by undoing the prediction and update step, but now based on the *quantized* values  $\hat{c}^j[n]$

$$\begin{aligned} \hat{c}_o^{j-1}[n] &= \hat{c}^{j-1}[2n+1] = \hat{d}[n] + P_{\hat{c}^j}^j(\hat{c}^j)[n], \\ \hat{c}_e^{j-1}[n] &= \hat{c}^{j-1}[2n] = 2\hat{c}^j[n] - \hat{c}^j[2n+1]. \end{aligned} \quad (16)$$

We now can compute the high-pass coefficients on the next finer level. By basing our choice of predictor at each stage on the quantized values  $\hat{c}$ , we maintain synchronization between encoder and decoder, and prevent propagation of quantization errors due to incorrect prediction filter choices. The encoder and decoder only need to agree on the level of quantization required for these choices.

Note that it is possible to use quantized data, not only for determining the prediction filter, but for the actual prediction as well. The lifting construction provides perfect reconstruction despite the presence of this nonlinear (quantization) prediction operator. The decoder and encoder will be synchronized not just in the *choice* of prediction filter, but also in the *output* of the prediction filter. However, the quality of this output (accuracy of the prediction) will be highly dependent on the level of quantization. Even for moderately quantized data, our research has shown that prediction errors will tend to be large, regardless of prediction filter. Thus, feeding quantized data into the prediction operator decreases the energy compaction properties of the wavelet transform and reduces the compression potential of our adaptive algorithm.

Finally, many modern image compression algorithms exploit the multiscale properties of the wavelet transform to provide embedded quantization. Our proposed scheme requires that the encoder and decoder agree on a minimum quantization level needed to make predictions decisions. Quantizing to a level below this minimum will lead to incorrect predictor choices and serious reconstruction error. However, using data quantized more accurately than the minimum level is still possible, since the encoder and decoder can always recreate the more coarsely quantized data required for the prediction decisions. Thus, embedded transmission is possible beyond the minimum quantization level required for the encoder/decoder synchronization.

### D. 2-D Prediction Windows

Since all the quantized coarse coefficients are available to both the encoder and decoder, we can utilize the data above and below the point of interest to determine our choice of predictors. That is, our edge-detection algorithm can analyze the data in this nonseparable 2-D prediction window to determine the location and orientation of the edge. Edges in images are actually contours; they have significant geometric structure. By using a nonseparable 2-D window, we can exploit this edge structure to make smarter prediction decisions within the framework of our separable transform.

If we sense (by our outlier method described earlier) that an edge is present within our prediction window, we analyze the data in the 2-D window around the point of interest to refine our estimate of the edge. We assume a step edge is present, project the data onto a truncated Fourier basis, and qualify our projection against our edge model. This process is a modified version of the algorithm presented in [24]. If the data passes our edge criteria, the intensity and angle of the edge are determined. This information is then used to refine our choice of prediction filter.

It is also possible to use not only the low-pass coefficients for prediction of  $x_o[n]$ , but also other odd coefficients in a causal neighborhood of  $x_o[n]$ . Suppose our signal  $x[n]$  is a row in an image. We would predict  $x_o[n]$  from low-pass coefficients  $c[n]$  on its left and right. Further suppose we have discovered a vertical step edge near  $x_o[n]$ . The precise location of the edge cannot be determined from the low-pass coefficients  $c[n]$ . However, if we know the value of the coefficients from the row *directly above*  $x_o[n]$ , we can use this information in the

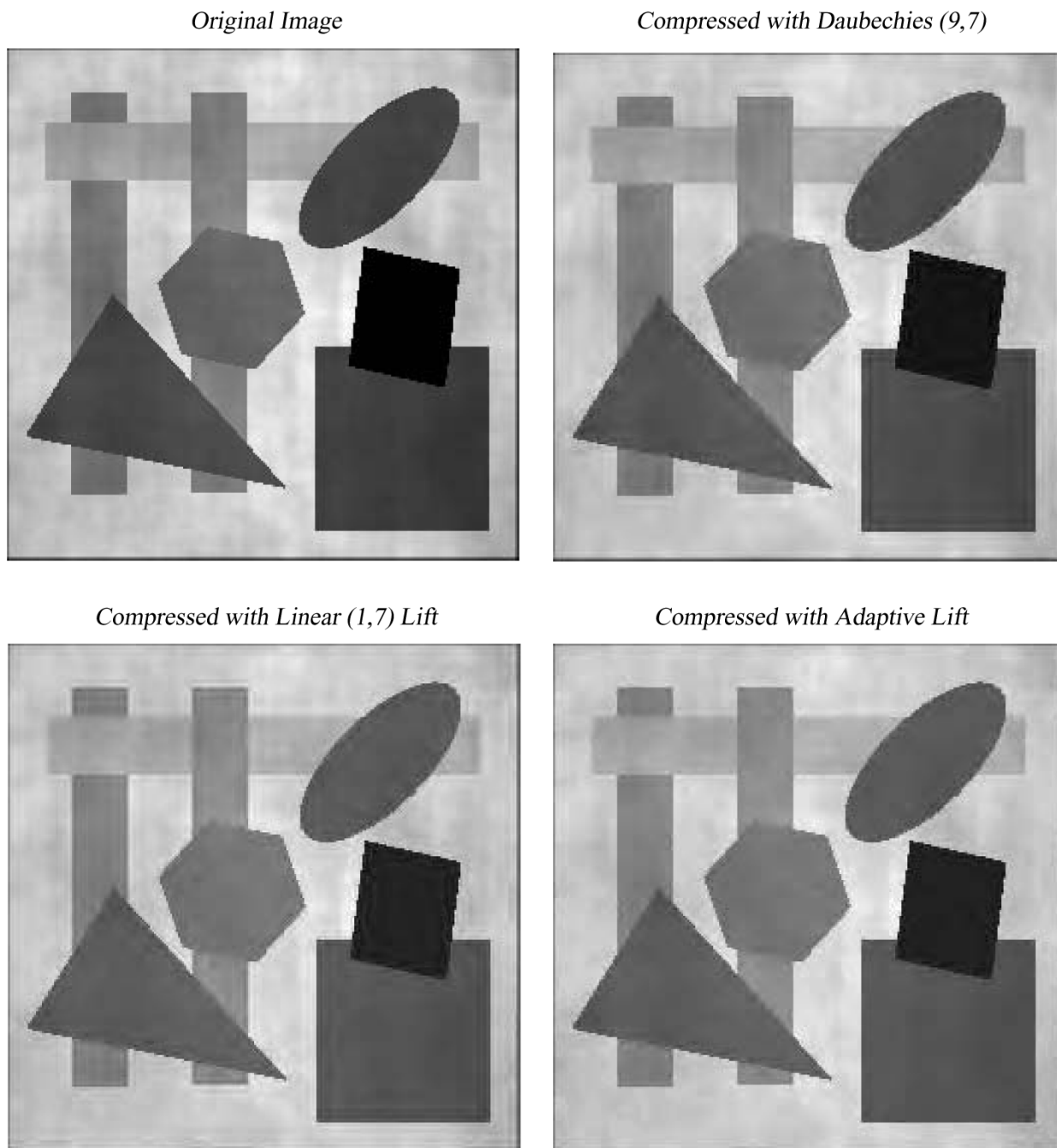


Fig. 10. Edge dominated image with texture, compressed to 0.67 BPP (12:1 compression). Note the ringing around the edges of the square in the Daubechies (9,7) and linear (1,7) lift images that is eliminated by the adaptive lift.

prediction of  $x_o[n]$ . This predict-from-above idea is similar to the causal  $P_b$  filter of the S+P algorithm [16] discussed in Section II-C.

Unfortunately, the predict-from-above scheme typically results in decreased stability. Consider the example above in which we resolve difficulties in predicting the location of a vertical edge in a row of coefficients by using already inverted coefficients in the row above. Such a scheme permits a quantization error in one row to propagate along a vertical edge to all other rows. We can prevent such propagation by employing a Differential Pulse Code Modulation (DPCM)-like strategy [25] of using *quantized* data from the causal neighbor-

hood for making predictions in the encoder as well as in the decoder. Note that this strategy will only work in the horizontal direction, since the vertical transform must be completed before the horizontal transform can be computed. Also, the quality of this predict-from-above will be highly dependent on the level of quantization. Even for moderately quantized data, the prediction errors tend to be large, again decreasing the energy compaction and compression potential. Thus, using the quantized data in this 2-D window to perform the actual prediction was found to decrease performance, and is therefore not included in our adaptive lifted algorithm. However, the nonseparable 2-D prediction window leads to a better choice of

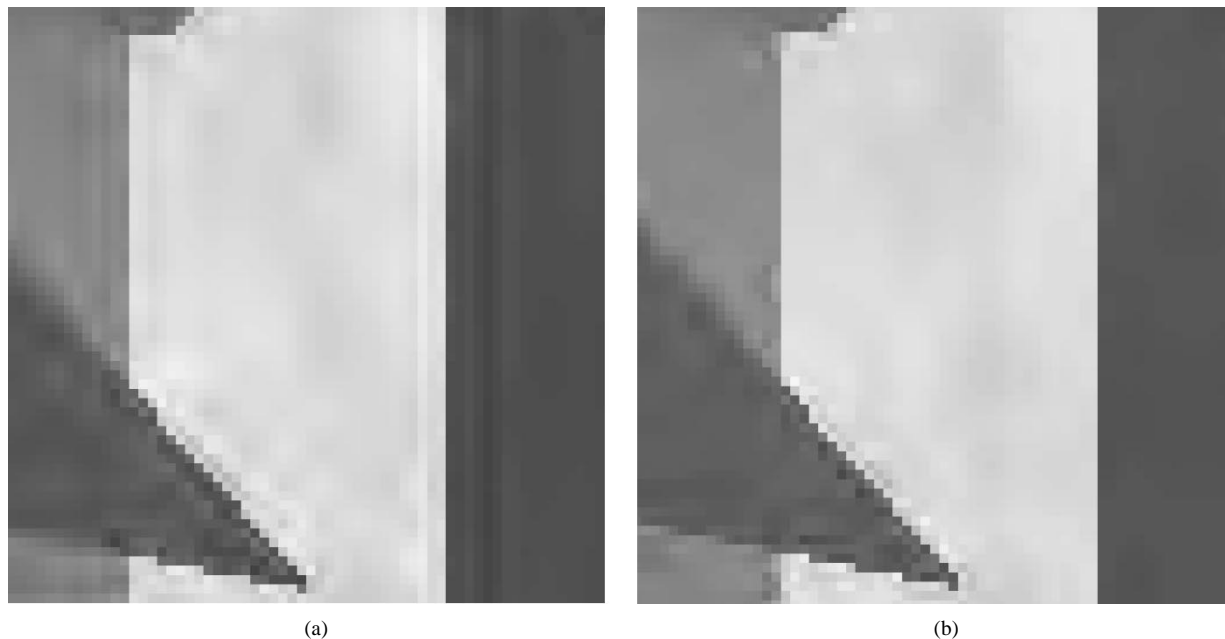


Fig. 11. Close-up of edge dominated image with texture, compressed to 0.67 bits-per-pixel (BPP) (12:1 compression). Note the sharp edges and reduced ringing with the adaptive algorithm. (a) Compressed with Daubechies (9,7). (b) Compressed with adaptive lift.

prediction filters (and better compression), and is an important part of our transform.

## V. RESULTS

### A. Synthetic Data

Fig. 10 shows the result of our adaptive lifting algorithm applied to an edge-dominated test image. This image was constructed by superimposing texture on shapes of different magnitudes and orientations. The original image was transformed and compressed to 0.67 bits-per-pixel (BPP) (12:1 compression) using an embedded zero-tree encoder [26]. For simplicity, we compress the zero-tree symbol stream with a Huffman coder, and we make no effort to compress the quantization bit stream. We compare our performance against that of the Daubechies (9,7) wavelet because it is very popular in image compression. We also compare against the linear (1,7) lift; it is the smoothest member of the family of wavelets that we use in our adaptive lifting algorithm.

We notice that the Daubechies (9,7) and linear (1,7) lift transformed images suffer from blurring and ringing around the edges. However, the image transformed with our adaptive lifted algorithm has much sharper edges. Ringing is reduced, edge sharpness is maintained, and the background texture is not significantly corrupted. These improvements are very visible in the closeup shown in Fig. 11. The reason for these improvements is that edges in our new transform are represented in a more compact fashion, and as a result there is less degradation of the image when we zero out small, nonzero coefficients.

As a performance metric, we compute the peak signal to noise ratio (PSNR),

$$\text{PSNR} = 20 \log \left( \frac{\max(x_i)}{\sqrt{\frac{\sum (x_i - \hat{x}_i)^2}{M}}} \right),$$

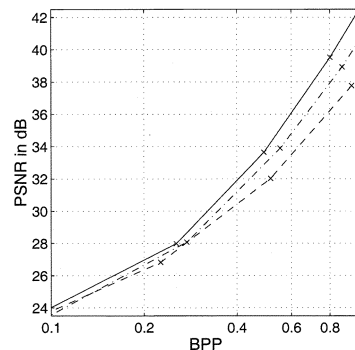


Fig. 12. Peak signal-to-noise ratio (PSNR) curves for the edge-dominated test image of Fig. 10. This test image was designed to demonstrate the potential gains of the adaptive lift. The adaptive algorithm (solid line) outperforms the Daubechies (9,7) transform (dash-dot) and the (1,7) linear lift (dash). The encoder and decoder were synchronized for the adaptive algorithm.

where  $x_i$  is the  $i^{\text{th}}$  pixel of our original image,  $\hat{x}_i$  is the  $i^{\text{th}}$  pixel of our reconstructed image, and  $M$  is the total number of pixels. The PSNR curve (Fig. 12) demonstrates that, for this edge-dominated test image, the adaptive algorithm has better PSNR performance than both the Daubechies (9,7) and linear (1,7) lift transforms. The Daubechies (9,7) PSNR curve is shown for reference only; our goal is to improve the performance of the linear (1,7) lift through adaptivity.

### B. Real Data

In Fig. 13, we see the result of our adaptive lifting algorithm on the image *cameraman*, compressed to 0.25 BPP (32:1 compression). Our prediction decisions are based on data quantized to 7 iterations of the zero-tree encoder to ensure decoder/encoder synchronization. While ringing has been reduced in the horizontal and vertical edges, there are still some ringing artifacts in the diagonal direction. The reason for these remaining





Fig. 13. Cameraman image compressed to 0.25 BPP (32:1 compression). (a) Cameraman image. (b) Compressed with Daubechies (9,7). (c) Compressed with linear (1,7) lift. (d) Compressed with adaptive lift.

artifacts is that we are using a separable transform in which we seek to avoid horizontal and vertical edges.

Note in Fig. 14 the PSNR performance of our adaptive algorithm over the linear (1,7) lift. Each point on the PSNR curve was generated with decoder/encoder synchronization, and reflects embedded transmission beyond this minimum quantization level. Again, the performance of the popular Daubechies (9,7) transform is shown for reference. Although our adaptive algorithm does not match the PSNR performance of the Daubechies (9,7) transform, the visual quality of our algorithm is comparable, due to the reduction in edge artifacts. In general the adaptive algorithm results in much sharper decoded images. We conjecture that introducing adaptivity into the Daubechies (9,7) transform (an area of current research) would result in further PSNR increases.

## VI. CONCLUSIONS

Lifting provides insight into the construction of the wavelet transform, and allows us to incorporate adaptivity and nonlinear operators into the transform. We presented the *Update First* scheme to maintain control over the multi-resolution properties of the transform despite the presence of these nonlinearities.

Within this scheme, we introduced an algorithm that switches between various linear predictors to *avoid predicting across edges*. This algorithm efficiently represents edges and compacts energy into the lower subbands of the transform. In addition, we employed a *2-D nonseparable window* to make better predictor choices. The update-first scheme allowed us to make these improvements while maintaining synchronization between the encoder and decoder (to prevent propagation of quantization errors).

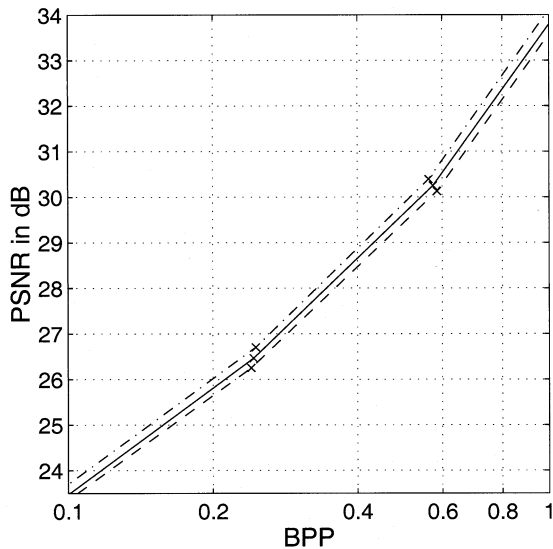


Fig. 14. PSNR curves for the cameraman image. The adaptive algorithm (solid line) outperforms its linear (1,7) lift (dash), but it does not meet the PSNR performance of the Daubechies (9,7) transform (dash-dot). However, edge artifacts are significantly reduced by the adaptive algorithm. The encoder and decoder were synchronized for the adaptive algorithm.

Our adaptive lifting transform appears promising for lossy compression. It reduces edge artifacts and ringing, and improves PSNR performance on certain test images.

Thus, the lifting scheme permits us to combine the best of both worlds. We can introduce nonlinear and adaptive filters into our transform, while simultaneously maintaining the multi-resolution properties of the linear wavelet transform. This provides a very powerful tool for not only lossy image compression, but other applications as well, such as lossless image compression [23], image estimation [27], image classification, and feature extraction.

#### ACKNOWLEDGMENT

G. Davis would like to acknowledge the hospitality of Rice University, Bell Laboratories, and Dartmouth College, where part of this work was done.

#### REFERENCES

- [1] D. Marr, *Vision*. New York: Freeman, 1982.
- [2] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: SIAM, 1992, vol. 61, CBMS-NSF Regional Conf. Series in Appl. Math..
- [3] R. J. Clark, *Transform Coding of Images*. New York: Academic, 1985.
- [4] R. Gonzalez and R. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.
- [5] R. Claypoole, G. Davis, W. Sweldens, and R. Baraniuk, "Nonlinear wavelet transforms image coding," in *Proc. Asilomar Conf. Signals, Systems, Computers*, Nov. 1997.
- [6] D. A. F. Florêncio and R. W. Schafer, "Perfect reconstruction nonlinear filter banks," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1996.
- [7] F. J. Hampson and J.-C. Pesquet, "A nonlinear subband decomposition with perfect reconstruction," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, 1996.
- [8] R. de Quieroz, D. A. F. Florêncio, and R. W. Schafer, "Non-expansive pyramid for image coding using a nonlinear filter bank," *IEEE Trans. Image Processing*, vol. 7, pp. 246–252, Feb. 1998.

- [9] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets," *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, 1996.
- [10] —, "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, vol. 29, no. 2, pp. 511–546, 1997.
- [11] E. Simoncelli, "Statistical models for images: Compression, enhancement and synthesis," in *Proc. Asilomar Conf. Signals, Systems, Computers*, 1997.
- [12] K. Ramchandran, S. LoPresto, and M. Orchard, "Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework," in *Proc. Data Compression Conf.*, Snowbird, UT, Mar. 1997.
- [13] C. Chrysafis and A. Ortega, "Efficient context-based entropy coding for lossy wavelet image compression," in *Proc. Data Compression Conference*, Mar. 1997, pp. 241–250.
- [14] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 245–267, 1998.
- [15] G. Deslauriers and S. Dubuc, "Symmetric iterative interpolation processes," *Constr. Approx.*, vol. 5, no. 1, pp. 49–68, 1989.
- [16] A. Said and W. A. Pearlman, "An image multiresolution representation for lossless and lossy image compression," *IEEE Trans. Image Processing*, vol. 5, no. 9, pp. 1303–1310, 1996.
- [17] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers," *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332–369, 1998.
- [18] P. J. Burt and E. H. Adelson, "Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, no. 4, pp. 532–540, 1983.
- [19] O. Gerek and E. Cetin, "Linear/nonlinear adaptive polyphase subband decomposition structures for image compression," in *IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3, 1998, pp. 1345–1348.
- [20] D. L. Donoho, "Smooth wavelet decompositions with blocky coefficient kernels," in *Recent Advances in Wavelet Analysis*: Academic Press, 1993, pp. 259–308.
- [21] W. Sweldens and P. Schröder, "Building your own wavelets at home," in *Wavelets in Computer Graphics*, 1996, ACM SIGGRAPH Course Notes, pp. 15–87.
- [22] A. Cohen, I. Daubechies, and J. Feauveau, "Bi-orthogonal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 45, pp. 485–560, 1992.
- [23] N. V. Boulgouris, D. Tzovaras, and M. G. Strintzis, "Lossless image compression based on optimal prediction, adaptive lifting and conditional arithmetic coding," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 1–14, 2001.
- [24] K. Jensen and D. Anastassiou, "Subpixel edge localization and the interpolation of still images," *IEEE Trans. Image Processing*, vol. 4, pp. 285–295, Mar. 1995.
- [25] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*. Englewood Cliffs, NJ: Prentice-Hall, 1984.
- [26] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.
- [27] R. Claypoole, R. Baraniuk, and R. Nowak, "Adaptive wavelet transforms via lifting," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, May 1998.



**Roger Claypoole** (M'93–SM'03) received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1989 and was commissioned into the United States Air Force. He received the M.S.E.E. degree from the Air Force Institute of Technology in 1994 and the Ph.D. degree from Rice University, Houston, TX, in 2000.

From 1990 to 1993, he was a Radar Engineer at the F-16 Systems Program Office. From 1995 to 1996, he was a Senior Spacecraft Engineer at the Space-Based Infrared Systems Program Office. Since 1999, he has been assigned to the Air Force Institute of Technology as an Assistant Professor of electrical engineering. His research interests include wavelet theory, image processing, and adaptive signal processing.



**Geoff Davis** received the Ph.D. degree in mathematics from New York University's Courant Institute of Mathematical Sciences in 1994.

From 1994 to 1998, he was a John Wesley Young Research Instructor and then an Assistant Professor with the Math Department at Dartmouth College. He spent the spring of 1997 visiting the Electrical and Computer Engineering Department, Rice University, Houston, TX, where the work in this paper was conceived. He subsequently joined the Signal Processing Group, Microsoft Research, Microsoft Corporation.

He is currently a freelance Software Developer based in Raleigh, NC.



**Wim Sweldens** (M'97) received the Ph.D. degree in applied mathematics in 1994 from the Katholieke Universiteit Leuven, Belgium.

He has been with Bell Laboratories, Lucent Technologies, Murray Hill, NJ, since 1995, where he is currently the Director of the Algorithm Research Department. His research is concerned with wavelets and multiscale analysis and its application in numerical analysis, signal processing, computer graphics, and wireless communications. He is the inventor of the lifting scheme, a new design and implementation

technique for wavelets that now is part of the JPEG2000 standard.

Dr. Sweldens was recently chosen one of the 100 most promising young innovators by MIT's *Technology Review*.



**Richard G. Baraniuk** (S'85-M'93-SM'98-F'02) received the B.Sc. degree in 1987 from the University of Manitoba, Manitoba, BC, Canada, the M.Sc. degree in 1988 from the University of Wisconsin, Madison, and the Ph.D. degree in 1992 from the University of Illinois at Urbana-Champaign, all in electrical engineering.

In 1986, he was a Research Engineer with Omron Tateisi Electronics, Kyoto, Japan. After spending 1992-1993 with the Signal Processing Laboratory of Ecole Normale Supérieure, Lyon, France, he joined

Rice University, Houston, TX, where he is currently a Professor of electrical and computer engineering. He spent Autumn 1998 at the Isaac Newton Institute of Cambridge University, Cambridge, U.K., as the Rosenbaum Fellow and 2001-2002 on sabbatical at Ecole Nationale Supérieure de Telecommunications in Paris and Ecole Fédérale Polytechnique de Lausanne in Switzerland. His research interests lie in the area of signal and image processing and include wavelets, probabilistic models, networks, and time-frequency analysis. He serves on the editorial board of *Applied and Computational Harmonic Analysis*.

Dr. Baraniuk received a NATO postdoctoral fellowship from NSERC in 1992, the National Young Investigator award from the National Science Foundation in 1994, a Young Investigator Award from the Office of Naval Research in 1995, the Rosenbaum Fellowship from the Newton Institute in 1998, the C. Holmes MacDonald National Outstanding Teaching Award from Eta Kappa Nu in 1999, the Charles Duncan Junior Faculty Achievement Award from Rice in 2000, the ECE Young Alumni Achievement Award from the University of Illinois in 2000, and the George R. Brown Award for Superior Teaching at Rice in 2001 and 2003. He was co-author on a paper with M. Grouse and R. Nowak that received the IEEE Signal Processing Society Junior Paper Award in 2001. He sits on the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society.