The Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Constructions

Wim Sweldens

Katholieke Universiteit Leuven Belgium, Department of Computer Science

Wim.Sweldens@cs.kuleuven.ac.be

ABSTRACT

In this paper we present the basic idea behind the lifting scheme, a new construction of biorthogonal wavelets which does not use the Fourier transform. In contrast with earlier papers we introduce lifting purely from a wavelet transform point of view and only consider the wavelet basis functions in a later stage. We show how lifting leads to a faster, fully in-place implementation of the wavelet transform. Moreover, it can be used in the construction of second generation wavelets, wavelets that are not necessarily translates and dilates of one function. A typical example of the latter are wavelets on the sphere.

Keywords: wavelet, biorthogonal, in-place calculation, lifting

1 Introduction

At the present day it has become virtually impossible to give the definition of a "wavelet". The research field is growing so fast and novel contributions are made at such a rate that even if one manages to give a definition today, it might be obsolete tomorrow. One, very vague, way of thinking about wavelets could be:

"Wavelet are **building blocks** *that can* **quickly decorrelate** *data."*

This sentence at least incorporates three of the main features of wavelets. First of all, they are *building blocks* for general data sets or functions. Mathematically we say that they form a basis or, more general a frame. This means that each element of a general class can be written in a stable way as a linear combination of the wavelets. If we denote the wavelets by ψ_i and the coefficients by γ_i, we can write a general function f as

$$f = \sum_i \gamma_i \psi_i.$$

Secondly, wavelets have the power to *decorrelate*. This means that the representation of the data in terms of

the wavelet coefficients γ_i is somehow more "compact" then the original representation. In information-theoretic jargon, we say that the entropy in the wavelet representation is smaller then in the original representation. In approximation-theoretic jargon, we want to get an accurate approximation of f by only using a small fraction of the wavelet coefficients.

The way to get this decorrelation power is to construct wavelets which already in some way resemble the data we want to represent. More specifically, we would like the wavelets to have the same correlation structure as the data. For example, most signals we encounter in daily life have both correlation in space and frequency. Samples which are spatically close are much more correlated then ones that are far apart, and frequencies often occur in bands. To analyze and represent such signals we need wavelets that are local in space and frequency. Typically this is achieved by building wavelets which have compact support (localization in space), which are smooth (decay towards high frequencies), and which have vanishing moments (decay towards low frequencies).

Finally, we want to *quickly* find the wavelet representation of the data. More precisely, we want to switch between the original representation of the data and its wavelet representation in a time proportional to the size of the data. The fast decorrelation power of wavelets is the key to applications such as data compression, fast data transmission, noise cancellation, signal recovering, and fast numerical algorithms.

The purpose of this paper is to introduce the *lifting scheme*, a new tool in the construction of biorthogonal wavelets. The main difference with classic constructions such as[1–3] is that it does not employ the Fourier transform. Until recently, the Fourier transform has been instrumental in wavelet constructions. The underlying reason is that wavelets are traditionally defined as translates and dilates of one function, and translation and dilation become algebraic operations after Fourier transform. The wavelet construction then relies on certain polynomial factorizations. We refer to wavelets which are translates and dilates of one function as *first generation wavelets*. In the case of first generation wavelets, the lifting scheme will never come up with wavelets which somehow could not be found by the techniques developed by Cohen, Daubechies, and Feauveau in.[2] Nevertheless, using lifting to (re)construct these wavelets has the following advantages:

1. It allows a faster implementation of the wavelet transform. Traditionally, the fast wavelet transform is calculated with a two-band subband transform scheme. In each step the signal is split into a high pass and low pass band and then subsampled. Recursion occurs on the low pass band. The lifting scheme makes optimal use of similarities between the high and low pass filters to speed up the calculation. In some cases the number of operations can be reduced by a factor of two.
2. The lifting scheme allows a fully in-place calculation of the wavelet transform. In other words, no auxiliary memory is needed and the original signal (image) can be replaced with its wavelet transform.
3. In the classical case, it is not immediately clear that the inverse wavelet transform actually is the inverse of the forward transform. Only with the Fourier transform one can convince oneself of the perfect reconstruction property. With the lifting scheme, the inverse wavelet transform can immediately be found by undoing the operations of the forward transform. In practise, this comes down to simply reversing the order of the operations and changing each $+$ into a $-$ and vice versa.
4. The lifting scheme is a very natural way to introduce wavelets in a classroom. Indeed, since it does not rely on the Fourier transform, the properties of the wavelets and the wavelet transform do not appear as

somehow "magical" to students who do not have a strong background in Fourier analysis.

Since lifting does not rely on the Fourier transform, it can be used to construction wavelets in settings where translation and dilation, and thus the Fourier transform, cannot be used. We refer to such wavelet as *second generation wavelets*. Typical examples are:

1. *Wavelets on bounded domains*: The construction of wavelets on domains in a Euclidean space is needed in applications such as image segmentation and the numerical solution of partial differential equations. A special case is the construction of wavelets on an interval, which is needed to transform finite length signals without introducing artifacts at the boundaries.
2. *Wavelets on curves and surfaces*: To analyze data that live on curves or surfaces or to solve equations on curves or surfaces, one needs wavelets intrinsically defined on these manifolds, independent of parametrization.
3. *Weighted wavelets*: Diagonalization of differential operators and weighted approximation require a basis adapted to weighted measures. Wavelets biorthogonal with respect to a weighted inner product are needed.
4. *Wavelets and irregular sampling*: Many real life problems require basis functions and transforms adapted to irregularly sampled data.

It is obvious that wavelets adapted to these setting cannot be formed by translation and dilation. The Fourier transform can thus no longer be used as a construction tool. The lifting scheme provides an alternative.

There are two ways to introduce lifting. The first one is concerned with the basis functions, i.e. the scaling functions, dual scaling functions, wavelets, and dual wavelets, and how lifting affects them. This approach was taken in the original papers.[9,10] In this paper, however, we follow a different approach, namely we first discuss how lifting affects the wavelet transform. We have found this to be a much more natural way to introduce lifting. In a later section, we will briefly mention what happens to the basis functions. Of course, theoretically both approaches are equivalent. In fact one can be seen as adjoint to the other.

2 The basic idea behind lifting

A canonical case of lifting consists of three stages, which we refer to as: *split*, *predict*, and *update*. We here describe the basic idea behind each and later work out a concrete example. Assume we start with an abstract data set, which we refer to as λ_0. We know this data set has some correlation structure and we would like to exploit it to obtain a more compact representation.

In the first stage we split the data into two smaller subsets λ_{-1} and γ_{-1}. (We use negative indices here because the convention is that the smaller the data set, the smaller the index.) We refer to γ_{-1} as the wavelet subset. We do not impose any no restriction on how the data should be split, nor on the relative size of each of the subsets. The only thing we need is some procedure to join λ_{-1} and γ_{-1} back into the original data set λ_0. The easiest

possibility for the split is a simply brutal cut of the data set into two disjoint parts. This choice we refer to as the *Lazy wavelet*. Think for example of cutting an image into two parts with a pair of scissors.

As we said before, we would like to get a more compact representation of λ_0. Consider the case were γ_{-1} does not contain any information (e.g. that part of the image is entirely black). Then we would immediately have a more compact representation since we can simply replace λ_0 with the smaller set λ_{-1}. Indeed the extra part needed to reassemble λ_0 does not contain any information.

Obviously this situation hardly ever occurs in practise. Therefore, in a second stage, we try to use the λ_{-1} subset to predict the γ_{-1} subset based on the correlation present in the original data. If we can find a prediction operator $\mathcal{P}$, independent of the data, so that

$$\gamma_{-1} \;=\; \mathcal{P}(\lambda_{-1}),$$

then again we can replace the original data set with λ_{-1}, since now we can predict the part missing to reassemble λ_0. The construction of a prediction operator is typically based on some model of the data which reflects its correlation structure. Obviusly the prediction operator $\mathcal{P}$ cannot be dependent on the data, otherwise we would hide information in $\mathcal{P}$.

Again, in practise it might not be possibly to exactly predict γ_{-1} based on λ_{-1}. However, $\mathcal{P}(\lambda_{-1})$ is likely to be close to γ_{-1}. Thus we might want to replace γ_{-1} with the difference between itself and its predicted value $\mathcal{P}(\lambda_{-1})$. If the prediction is reasonable, this difference will contain much less information then the original γ_{-1} set. We denote this abstract difference operator with a $-$ sign and thus get

$$\gamma_{-1} \;:=\; \gamma_{-1} - \mathcal{P}(\lambda_{-1}).$$

The wavelet subset now encodes how much the data deviates from the model on which $\mathcal{P}$ was built.

We now have some more insight in how to split the original data set. Indeed, in order to get the maximal data reduction from prediction, we need the subsets λ_{-1} and γ_{-1} to be maximally correlated. Cutting an image into a left and right part might not be the best idea since pixels on the far left and the far right are hardly correlated. Predicting the right half of the image based on the left is thus a though job. A better idea is to interlace the two sets. We will come back to this later.

At this moment we can replace the original data with the smaller set λ_{-1} and the wavelet set γ_{-1}. With a good prediction, the two subsets $\{\lambda_{-1}, \gamma_{-1}\}$ yield a more compact representation then the orginal set λ_0. We can now iterate this scheme. We split λ_{-1} into two subsets λ_{-2} and γ_{-2} and then replace γ_{-2} with the difference between γ_{-2} and $\mathcal{P}(\lambda_{-2})$. After n steps we have replaced the original data with the wavelet representation $\{\lambda_{-n}, \gamma_{-n}, \cdots, \gamma_{-1}\}$. Given that the wavelet sets encode the difference with some predicted value based on a correlation model, this is likely to give a more compact representation.

This scheme sounds promising, but in some cases we are not completely satisfied. The reason is that we often want some global properties of the original data set to be maintained in the smaller versions λ_{-j}. For example, in the case of an image, we would like the smaller images λ_{-j} to have the same overall brightness, i.e. the same

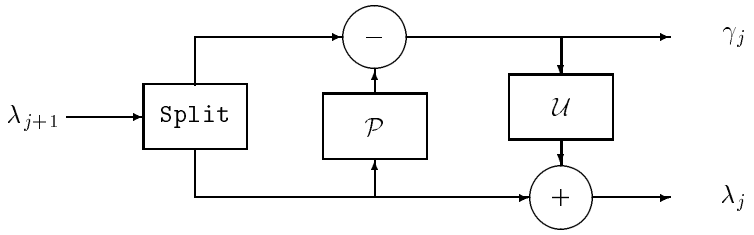

Figure 1: *The lifting scheme: split, predict, and update.*

average pixel value. If the splitting stage is simply subsampling and we iterate the scheme till λ_{-n} is only 1 pixel, that pixel will be an arbitrary pixel from the original image. We would rather have the last value to be the average of all the pixel values in the original image. In fact, we are facing a worst case example of a problem known as aliasing.

We can solve part of this problem by introducing a third stage. The idea is to find a better λ_{-1} so that a certain scalar quantity $Q()$, like e.g. the mean, is preserved, or

$$Q(\lambda_{-1}) \;=\; Q(\lambda_0).$$

We could do so by finding a new operator to extract λ_{-1} directly from λ_0, but we decide not to for two reasons. First, this would create a scheme which is very hard to invert. Secondly, we would like to reuse the work already done maximally. Therefore we propose to use the already computed wavelet set γ_{-1} to update λ_{-1} so that the latter preserves $Q()$. In other words, we construct an operator $\mathcal{U}$ and update λ_{-1} as

$$\lambda_{-1} \;:=\; \lambda_{-1} + \mathcal{U}(\gamma_{-1}).$$

The three stages of lifting are depicted in a block diagram in Figure 1. Again we can now iterate the scheme. This leads to the following wavelet transform algorithm (with a C-like syntax):

$$\texttt{For j = -1 downto -n:} \quad \left\{ \begin{array}{rcl} \{\lambda_j, \gamma_j\} & := & \texttt{Split}(\lambda_{j+1}) \\ \gamma_j & -= & \mathcal{P}(\lambda_j) \\ \lambda_j & += & \mathcal{U}(\gamma_j)\,. \end{array} \right.$$

We can now illustrate one of the nice properties of lifting: once we have the forward transform, we can immediately derive the inverse. The only thing to do is to reverse the operations and toggle $+$ and $-$. This leads to the following algorithm for the inverse wavelet transform:

$$\texttt{For j = -n to -1:} \left\{ \begin{array}{rcl} \lambda_j & -= & \mathcal{U}(\gamma_j) \\ \gamma_j & += & \mathcal{P}(\lambda_j) \\ \lambda_{j+1} & := & \texttt{Join}(\lambda_j, \gamma_j)\,. \end{array} \right.$$

3 A simple example

In this section we consider a simple example to illustrate the ideas of the previous section. Suppose we sample a signal $f(t)$ with sampling distance $\Delta t = 1$. We denote the original samples by $\lambda_0 = \{\lambda_{0,k} = f(k) \mid k \in \mathbf{Z}\}$.

We first need to define the split stage. As mentioned in the previous section this implies splitting the data into two parts which are maximally correlated. As the correlation in most signals is local, i.e. neighboring samples are much more correlated then ones that are far apart, we simply subsample the data into even and odd indexed samples. This is an example of a Lazy wavelet transform. We obtain two sequences λ_{-1} and γ_{-1} with coefficients

$$\lambda_{-1,k} \;:=\; \lambda_{0,2k} \quad \text{and} \quad \gamma_{-1,k} \;:=\; \lambda_{0,2k+1} \quad \text{for} \quad k \in \mathbf{Z}\,. \tag{1}$$

Next we need to find a operator to predict γ_{-1} based on λ_{-1}. Again assuming maximal correlation amongst neighboring samples, we simply suggest to predict an odd sample $\lambda_{0,2k+1}$ as the average of its two (even) neighbors: $\lambda_{-1,k}$ and $\lambda_{-1,k+1}$. The difference with this prediction then becomes

$$\gamma_{-1,k}\;,:=\; \gamma_{-1,k} - 1/2\,(\lambda_{-1,k} + \lambda_{-1,k+1})\,. \tag{2}$$

The model used to build $\mathcal{P}$ is a function piecewise linear over intervals of length 2. If the original signal complies with the model, all wavelet coefficients in γ_{-1} are zero. In other words, the wavelet coefficients measure to which extend the original signal *fails to be linear*. Their expected value is small. In terms of frequency content, the wavelet coefficients capture high frequencies present in the original signal.

However, the frequency localization of the signals λ_{-1} and γ_{-1} is far from ideal. It would be nice if the λ_{-1} signal somehow captures the low frequencies, and the γ_{-1} the high frequencies. Right now the λ_{-1} signal is simply subsampled and its frequency content thus stretches out over the whole band of the original signal. Again, we have the worst case example of aliasing. In the update stage, we can reduce the amount of aliasing by at least assuring that the DC component ends up entirely in the λ_{-1} part. In other words, we would like the average of the signal to be maintained in λ_{-1}, or

$$\sum_k \lambda_{-1,k} \;=\; 1/2 \sum_k \lambda_{0,k}.$$

This is precisely the scalar quantity $Q()$ of the previous section which we would like to preserve. We therefore update the $\lambda_{-1,k}$ with the help of the wavelet coefficients $\gamma_{-1,k}$. Again we use the neighboring wavelet coefficients and thus propose an update $\mathcal{U}$ of the form:

$$\lambda_{-1,k} \;:=\; \lambda_{-1,k} + A\,(\gamma_{-1,k-1} + \gamma_{-1,k})\,.$$

To find A we calculate the average:

$$\sum_k \lambda_{-1,k} \;=\; \sum_k \lambda_{0,2k} + 2\,A \sum_k \gamma_{-1,k} \;=\; (1 - 2\,A) \sum \lambda_{0,2k} + 2\,A \sum \lambda_{0,2k+1}.$$

From this we see that the correct choice to maintain the average is $A = 1/4$. One step in the wavelet transform is shown in the scheme in Figure 2. By iterating this scheme we get a complete wavelet transform. The inverse transform can be derived immediately as shown in the previous section. Note that at no point we used the Fourier transform.

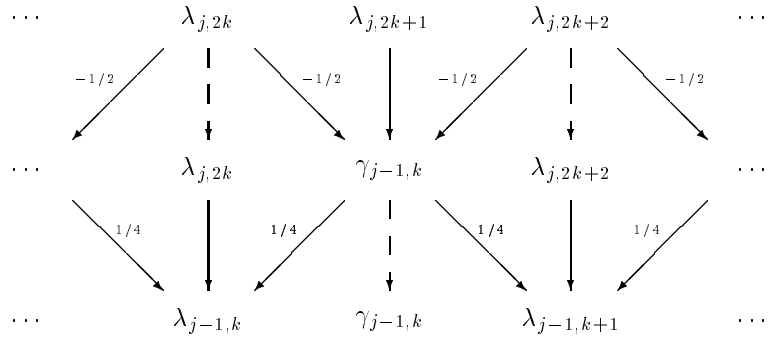

Figure 2: *The lifting scheme: Split, calculate the wavelet coefficients $\gamma_{j-1,m}$ as the failure to be linear, and use them to update the $\lambda_{j-1,k}$.*

The wavelet transform presented here in fact is the $(N = 2, \widetilde{N} = 2)$ biorthogonal wavelet transform of Cohen-Daubechies-Feauveau.[2] This simple example already shows how the lifting scheme can speed up the implementation of the wavelet transform. Classicly the $\{\lambda_{-1,k}\}$ coefficients are found as the convolution of the $\{\lambda_{0,k}\}$ coefficients with the filter $\widetilde{h} = \{-1/8, 1/4, 3/4, 1/4, -1/8\}$. This step would take 6 operations per coefficient while lifting only needs 3.

This is only one simple instance of lifting. A whole family of biorthogonal wavelets can be constructed by varying the three stages of lifting:

1. *Split:* Other choices but the Lazy wavelet are possible as an initial split. A typical alternative is the Haar wavelet transform.

2. *Predict:* In wavelet terminology the prediction step establishes the number of vanishing moments (N) of the dual wavelet. In other words, if the original signal is a polynomial of degree less than N, all wavelet coefficients will be zero. In our example $N = 2$, but higher order schemes are easily obtained by involving more neighbors.

3. *Update:* Again in wavelet terminology, the update step establishes the number of vanishing moments ($\widetilde{N}$) of the primal wavelet. In other words the transform preserves the first $\widetilde{N}$ moments of the λ_j sequences. The example above had $\widetilde{N} = 2$ (one extra because of symmetry). Again higher order ones can be constructed by involving more neighbors. In some cases, namely when the split stage already creates a wavelet with a vanishing moment (such as the Haar) the update stage can be omitted.

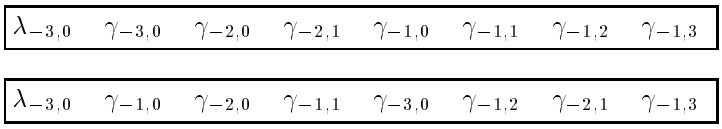

Figure 3: *Mallat (above) and Lifting (below) organization of wavelet coefficients for* $n = 3$.

4 In-place calculation

In this section we introduce another feature of lifting: in-place calculation. This means we can replace the original data set with its wavelet transform without having to allocate extra memory. As is well known, the FFT has a similar feature provided one starts with bit-reversing the original samples. The basic idea behind in-place calculation of the wavelet transform using lifting is quite similar.

Assume the original signal has length 2^n and the original samples are stored in a vector so that sample $\lambda_{0,k}$ sits in memory location k ($0 \leq k < 2^n$). In the traditional organization of the wavelet coefficients, as proposed by Mallat in,[6,5] a wavelet coefficient $\gamma_{j,k}$ is stored in location $2^{n+j} + k$, see Figure 3 for an example with $n = 3$. To obtain in-place calculation with lifting, we propose a different ordering of the wavelet coefficients. The idea is to store the coefficient $\gamma_{j,k}$ in location $2^{-j-1} + 2^{-j}\,k$, see Figure 3. Essentially all the coefficients depicted in one column of Figure 2 are stored in the same location. The Lazy wavelet transform is then immediate. Since all other operations can be done with $+=$ or $-=$ operations, we have a fully in-place calculation. Figure 4 shows the in-place organization of the wavelet coefficients of the classic Lena image.

It is also possible to first rearrange the original samples to end up with the Mallat organization of the coefficients after an in-place calculation. The sample $\lambda_{0,k}$ then has to be stored in location $m(k)$ which can be found as follows. Consider the n-bit binary representation of k: $\mathtt{k}_0\mathtt{k}_1\mathtt{k}_2\cdots\mathtt{k}_n$. Next isolate the trailing zeros (if any) as $\mathtt{k}_0\mathtt{k}_1\mathtt{k}_2\cdots\mathtt{k}_j100\ldots0$. Now the binary expansion of $m(k)$ is $00\cdots01\mathtt{k}_0\mathtt{k}_1\mathtt{k}_2\cdots\mathtt{k}_j$. In other words, it can be seen a partial bit reversal.

5 The basis functions

So far, we only explained how lifting affects the wavelet transform. In this section we briefly describe what happens to the basis functions. The basic idea to find a basis function given the transform is very simple. If you want to construct the basis function associated with a coefficient $\lambda_{j,k}$ of $\gamma_{j,k}$, simply put that coefficient to one, all other coefficients to zero, and perform an inverse wavelet transform starting from level j. This is known as the *cascade algorithm*.[4] We refer to the basis functions associated with the $\lambda_{j,k}$ (respectively $\gamma_{j,k}$) as scaling functions (respectively wavelets) and denote them with $\varphi_{j,k}$ (respectively $\psi_{j,k}$). If you are interested in the discrete basis, do an inverse transform till level 0, while if you are interested in the continuous basis, do an inverse transform add infinitum.

Figure 4: *In-place organization of Lena wavelet coefficients.*

This way it is easy to see that the Lazy wavelet simple corresponds to a Dirac sequence or a Dirac function. In case of the simple example mentioned earlier, the scaling functions are given by

$$\varphi_{j,k}(x) \;=\; \Lambda(2^j x - k),$$

where Λ is the classical "Hat" function: $\Lambda(x) = \max\{0, 1-|x|\}$. Essentially all scaling functions are translates and dilates of one particular function. The same is true for the wavelet. By doing one step of the inverse transform we see that

$$\psi_{j,k}(x) = \psi(2^j x - k)\,,$$

where $\psi(x)$ is given by

$$\psi(x) = \Lambda(2x-1) - 1/4\,\Lambda(x) - 1/4\,\Lambda(x+1)\,. \tag{3}$$

In case we would not have an update stage, the wavelet would simply be $\psi(x) = \Lambda(2x-1)$.

This gives us another insight into how lifting constructs wavelets. A new wavelet $\psi(x)$ is found as an old wavelet $\Lambda(2x-1)$ (the one without updating) combined with scaling functions on the *same* level, $\Lambda(x)$ and $\Lambda(x+1)$. This opposed to the classical case were a wavelet is constructed as a linear combination of Hat functions on the next *finer* level, namely $\Lambda(2x-k)$ with $k \in \mathbf{Z}$. The coefficients in (3) are chosen so that the wavelet has two vanishing moments. This is another way to get to the $1/4$ coefficients encountered earlier. The prediction stage actually corresponds to a similar operation on the dual wavelet, which is sometimes referred to as dual lifting. We thus start from an almost trivial case, the Lazy wavelet, and gradually build a new wavelet with improved properties, by adding in new basis functions. This is the inspiration behind the name "lifting scheme."

Let us now write the original signal as

$$f(x) \;=\; \sum_k \lambda_{0,k}\, \varphi_{0,k}\;.$$

After wavelet transform, the same signal can be written as

$$f(x) \;=\; \sum_k \lambda_{-n,k}\, \varphi_{-n,k} + \sum_{j=-n}^{-1} \sum_k \gamma_{j,k}\, \psi_{j,k}.$$

This is precisely a representation with "building blocks that decorrelate" which we were after in the introduction. Because of the correlation structure, many of the wavelet coefficients will be small. We can thus obtain an accurate approximation with only a small number of coefficients by simply omitting the wavelet coefficients below a certain threshold.

6 Second generation wavelets

The original motivation for the lifting was to construct wavelets in settings were no Fourier transform is available. The theory of lifting for second generation wavelets is given in.[10] Here we just introduce the basic idea.

The key point in each setting is to define the initial split operation. The easiest choice again is the Lazy wavelet transform. The prediction and updating stage are then quite similar to the ones described above. The main difference lies in the fact that the filter coefficients used in the prediction and update operator might vary depending on location. Indeed, if one wants to account for local irregularities, one cannot use the same filters everywhere. This is precisely why the Fourier transform can no longer be used. As a result the wavelets are not translates and dilates of one function. However, they still enjoy all the nice properties of first generation wavelets, such as fast transform and decorrelation power. They still are compactly supported, smooth, and have vanishing moments.

Let us first consider the case of wavelets on an interval. We essentially want to transform a signal of arbitrary finite length without the use of ad hoc solutions such as zero padding, periodization, or reflection around the edges. The Lazy wavelet transform can still be subsampling even and odd samples. In our simple example on the real line, the predicted value for an odd sample was based on its neighboring even samples left and right. In case of a finite length signal, the same idea can be used as long as the sample is sufficiently far away from the boundary. At the left boundary, if not enough even samples on the left are available to predict an odd sample, one simply replaces the missing samples on the left by extra samples on the right. In other words, we look for more correlated data where it is available. For example, suppose we use cubic interpolation to predict an odd sample based upon 4 neighboring even samples. Away from the boundary we use 2 samples on the left and 2 on the right. Close to the left boundary we might have to use 1 on the left and 3 on the right, or even none on the left and 4 on the right. This automatically leads to filters adapted for boundary constructions. It assures that all wavelets, including the ones at the boundary have the same number of vanishing moments. The wavelet coefficients for the Lena image in Figure 4 were calculated this way. Examples of boundary wavelets can be found in.[11]

In the case of irregular samples the same philosophy can be used. The fact that the sample locations are not on a regular grid poses no problems for the local polynomial prediction or update. The filters now change everywhere to account for the different locations of the samples. Here one has several choices for the Lazy wavelet. One idea is to still use even and odd subsampling. This way the imbalances in the sampling distances are maintained throughout the hierarchy. The alternative is to subsample in such a manner that the ratio of the largest versus smallest sampling distance approaches one. Which one is better depends on the situation at hand. For practical examples we again refer to.[11]

Another typical example of second generation wavelets are wavelets defined on curves, surfaces, or general manifolds. More particularly, the lifting scheme was used to construct wavelets on a sphere in.[7] These spherical wavelets are used for the efficient representation of data that naturally lives on a sphere. Examples are topographic data (earth elevation), bidirectional reflection functions, astrophysical data, and environment maps. These wavelets were used for spherical image processing in.[8]

7 Future developments

In this paper we gave a short introduction to lifting, a new method to construct wavelets. For more details and applications we need to refer to the original papers. Here we just would like to mention a few developments which are currently under investigation.

- Wavelets on general surfaces: The construction of the spherical wavelets does not fundamentally rely on the special properties of the sphere. It only uses the fact that one can recursively cut the sphere into spherical triangles. Therefore the construction can be generalized to more general surfaces or manifolds.
- Wavelet packets: The lifting scheme can also be used in the construction of wavelet packets. It is not hard to come up with a Lazy wavelet packet, i.e. a transform which also recursively splits the γ_{-j} sets. On these splittings again the prediction and update operators can be used.
- M-band wavelets: Again it is not very difficult to invent a Lazy M-band wavelet. Now we need to find several prediction and update operators.
- Wavelet frames: The lifting scheme can be used to construct overcomplete representations or frames. The key again lies in the correct definition of the Lazy wavelet. The two sets γ_{-1} and λ_{-1} coming from the split can have some amount of overlapping information or redundancy. Prediction and updating then would lead to wavelet frames.

Acknowledgment

The author is Senior Research Assistant of the National Fund of Scientific Research Belgium (NFWO). Part of this work was done at the University of South Carolina with support from NSF EPSCoR Grant EHR 9108772

and DARPA Grant AFOSR F49620-93-1-0083. The author would also like to acknowledge Interval Research in Palo Alto where he was visiting when writing this paper. Also special thanks to Peter Schröder for many most inspiring and stimulating discussions. The code to generate the in-place coefficients of the Lena image was written by two South Carolina graduate students, Gabriel Fernández and Senthil Periaswamy, who the author had the pleasure of working with in CS564 and CS798. Finally, it was Geoff Davis who pointed out that the in-place organization of the wavelet coefficients actually corresponds to a partial bit reversal of the indices.

8 REFERENCES

1. C. K. Chui. *An Introduction to Wavelets.* Academic Press, San Diego, CA, 1992.
2. A. Cohen, I. Daubechies, and J. Feauveau. Bi-orthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 45:485–560, 1992.
3. I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:909–996, 1988.
4. I. Daubechies. *Ten Lectures on Wavelets.* CBMS-NSF Regional Conf. Series in Appl. Math., Vol. 61. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992.
5. S. G. Mallat. Multifrequency channel decompositions of images and wavelet models. *IEEE Trans. Acoust. Speech Signal Process.*, 37(12):2091–2110, 1989.
6. S. G. Mallat. Multiresolution approximations and wavelet orthonormal bases of $L_2(\mathbf{R})$. *Trans. Amer. Math. Soc.*, 315(1):69–87, 1989.
7. P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics, (SIGGRAPH '95 Proceedings)*, 1995.
8. P. Schröder and W. Sweldens. Spherical wavelets: Texture processing. In P. Hanrahan and W. Purgathofer, editors, *Rendering Techniques '95.* Springer Verlag, Wien, New York, August 1995.
9. W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. Technical Report 1994:7, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1994.
10. W. Sweldens. The lifting scheme: A construction of second generation wavelets. Technical Report 1995:6, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995.
11. W. Sweldens and P. Schröder. Building your own wavelets at home. Technical Report 1995:5, Industrial Mathematics Initiative, Department of Mathematics, University of South Carolina, 1995.

Notes:

The lifting scheme has connections with many other developments. Space does not allow us to mention them here. They are all pointed out in the original papers, which leads to a bibliography of over a 100 items!

The technical reports from South Carolina are available as Postscript files through anonymous ftp to `ftp.math.sc.edu`, in the directories `/pub/imi_94` and `/pub/imi_95`.