Proceed. KVIV
Wavelet analysis : a new tool in signal and image analysis KVIV

Applying Integer Wavelet Transforms

Geert Uytterhoeven* Dirk Roose* Adhemar Bultheel*

December 11, 1996

Abstract

Classical wavelet transforms convert floating point numbers to floating point numbers. However, in many multimedia applications (images, video, audio) the input data consist of integer values only. Recent research produced new methods, based on the Lifting Scheme, that allow to perform all calculations in integer math. This also has important implications for hardware implementations.

1 Introduction

Classical wavelet transforms convert floating point numbers to floating point numbers. However, in many multimedia applications (images, video, audio) the input data consist of integer values only. In this paper we show how to implement integer wavelet transforms in an efficient way.

First we give a catalog of wavelet related terminology by reviewing the theory of the wavelet transform in one and two variables. The compression properties of the biorthogonal Cohen-Daubechies-Feauveau (CDF) wavelets are illustrated with a simple compression technique.

Next we discuss the Lifting Scheme into more detail. This scheme is an algorithm, originally designed to compute second generation wavelets in an efficient way. Later it has been shown that it can be used also to generate the "classical" (bi)orthogonal wavelets of Cohen-Daubechies-Feauveau (CDF) type, with several advantages over the classical computational schemes. We give an inventory of the lifting steps for some CDF wavelet transforms.

An advantage of the Lifting Scheme is that it can be converted easily into a transform that maps integers to integers, while retaining the perfect reconstruction property. We discuss how to implement the integer Lifting Scheme for CDF wavelets.

*Department of Computer Science, Katholieke Universiteit Leuven, Belgium. This research is supported by the Flemish Information Technology Action Program ('Vlaams Informatietechnologie Actieprogramma'), project number ITA/950244.

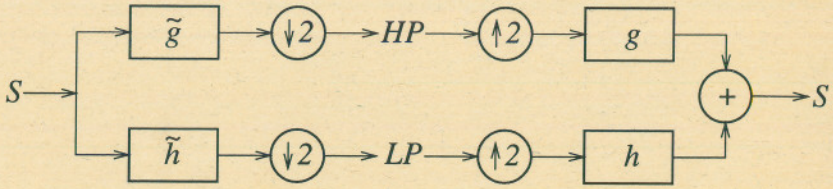


Figure 1: The filter bank algorithm.

2 The Wavelet Transform

2.1 The Basic Transform Step

The basic block in a wavelet transform [4] is a filter bank, consisting of 2 filters (cfr. fig. 1). A discrete signal S is filtered by a high pass filter ($\tilde{g}$) and a low pass filter ($\tilde{h}$), and downsampled. The results are a high pass (HP) and a low pass (LP) signal, each containing half as much samples as the input signal S.

For the inverse transform, first the signals HP and LP are upsampled by putting zeroes in between every sample. After that they are filtered by the filters g and h and the result is added together. In the case of perfect reconstruction — we'll only consider perfect reconstruction filter banks here — the resulting signal is equal to the original signal S.

The filters g, h, $\tilde{g}$ and $\tilde{h}$ are called wavelet filters if they fulfill certain conditions. The filters $\tilde{g}$ and $\tilde{h}$ are called dual filters w.r.t. g and h.

2.2 Wavelet and Scaling Functions

The wavelet filters g and h uniquely define a (primal) wavelet function $\psi(x)$ and a (primal) scaling function $\varphi(x)$. The shape of these functions can easily be found be iteratively applying an upsampling and filter step on a signal sequence containing only zeros, except for one sample. Analogously, the filters $\tilde{g}$ and $\tilde{h}$ define a dual wavelet function $\tilde{\psi}(x)$ and a dual scaling function $\tilde{\varphi}(x)$.

If the filters g and $\tilde{g}$, and h and $\tilde{h}$ are equal to each other, then the primal and dual wavelet and scaling functions coincide too. In this case the wavelets are called orthogonal wavelets. In the other, more generic, case they are called biorthogonal wavelets.

The regularity of a basis function can be measured by the number of vanishing moments. A function has n vanishing moments if the first n moments $\mathcal{M}_j$ ($j \leq n$)

$$\mathcal{M}_j = \int_{-\infty}^{\infty} x^j f(x)\,dx$$

are zero. The more vanishing moments a function has, the smoother the function is.

The mathematical meaning is that all polynomials x^j with a degree $j < n$ can be represented exactly in the basis $f_i(x)$ derived from the function $f(x)$:

$$x^j = \sum_i a_i f_i(x).$$

2.3 Example: Biorthogonal Cohen-Daubechies-Feauveau

This family of biorthogonal wavelets [2] has some interesting properties:

- The scaling function $\varphi(x)$ is always symmetric.
- The wavelet function $\psi(x)$ is always symmetric or antisymmetric.
- The wavelet filters are finite.
- The coefficients of the wavelet filters are of the form $\frac{z}{2^n}$, with z integer and n a natural number.

These wavelets are commonly classified by the number of vanishing moments they have: $(n, \tilde{n})$ means that the primal wavelet $\psi(x)$ — the synthesizing high-pass filter — has n vanishing moments, while the dual wavelet $\tilde{\psi}(x)$ — the analyzing high-pass filter — has $\tilde{n}$ vanishing moments.

The plots of some of the scaling functions $\varphi(x)$ and wavelet functions $\psi(x)$ of this family are shown in figure 2.

2.4 Multi-Resolution

The signal LP obtained by filtering with filter h and subsampling resembles the original signal S very much: it's a representation of S at a lower resolution level. On the contrary, HP contains the detail information that's lost by making the transition to that lower resolution level [5].

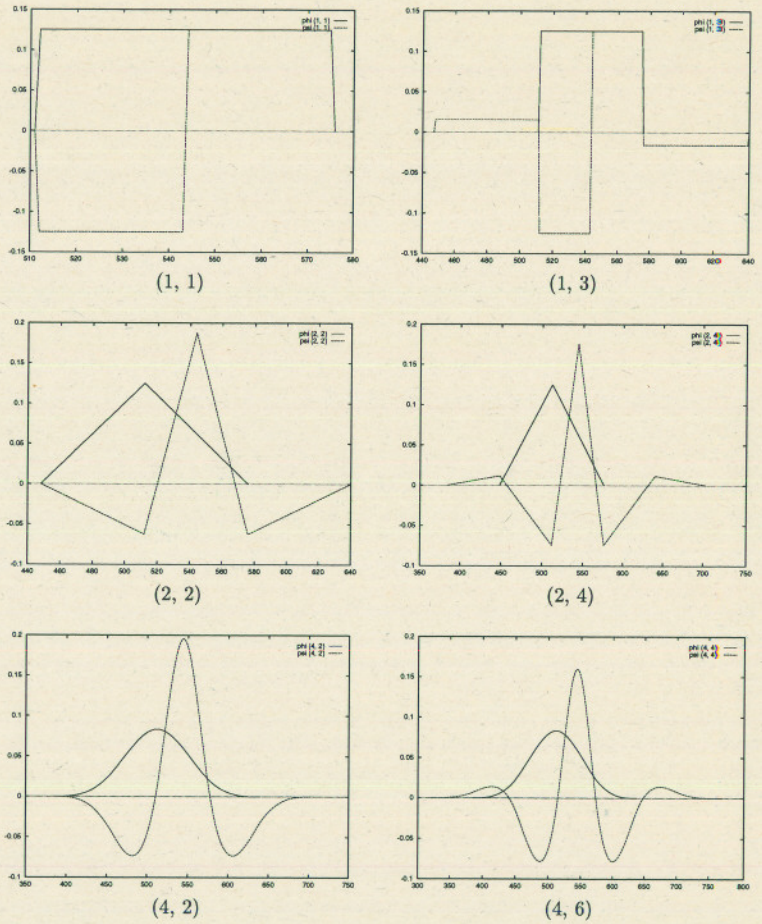
If we treat the low frequency information as a new signal and pass it through the same filter bank, and repeat these steps several times, we call this a wavelet transform. After some iterations we retain a very low frequency signal together with the detail information for the different resolution levels. This is shown by the decomposition tree in fig. 3. The blocks marked with V_n represent the low frequency signal at resolution level n, while the W_n represent the detail signal for the transition from resolution level n to level $n+1$.

Because of the subsampling the length of the low pass signal is divided by two after every filter step. Thus the wavelet transform has a linear complexity ($\mathcal{O}(n)$) and can be executed fast.

2.5 The Mathematics behind the Wavelet Transform

The scaling function $\varphi(x)$ and the wavelet function $\psi(x)$ and their duals $\tilde{\varphi}(x)$ and $\tilde{\psi}(x)$ satisfy dilation equations which are:

$$\begin{aligned}
\varphi(x) &= \sqrt{2}\sum_k h_k\,\varphi(2x-k),\\
\psi(x) &= \sqrt{2}\sum_k g_k\,\varphi(2x-k),\\
\tilde{\varphi}(x) &= \sqrt{2}\sum_k \tilde{h}_k\,\tilde{\varphi}(2x-k),
\end{aligned}$$



(1, 1) (1, 3)

(2, 2) (2, 4)

(4, 2) (4, 6)

Figure 2: Cohen-Daubechies-Feauveau biorthogonal wavelets.

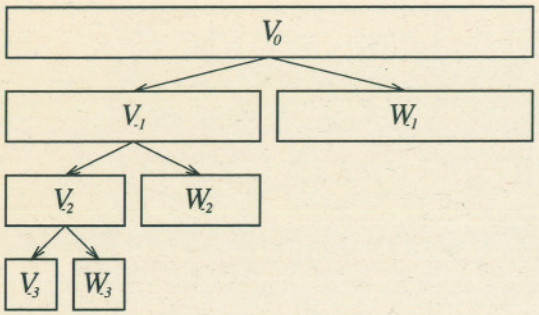


Figure 3: The decomposition tree for a wavelet transform.

$$\tilde{\psi}(x) = \sqrt{2}\sum_k \tilde{g}_k\,\tilde{\varphi}(2x-k).$$

The coefficients h_k, g_k, $\tilde{h}_k$ and $\tilde{g}_k$ represent the coefficients of the filters h, g, $\tilde{h}$ and $\tilde{g}$ respectively.

We define the basis functions

$$\begin{aligned}\varphi_{j,k}(x) &\equiv 2^{2/j}\varphi(2^j x-k),\\ \psi_{j,k}(x) &\equiv 2^{2/j}\psi(2^j x-k),\end{aligned}$$

with $j,k \in \mathbf{Z}$, i.e. $\varphi_{j,k}(x)$ is a translated and dilated version of $\varphi(x)$.

If we have a representation of a function at resolution level $j+1$:

$$f(x) = \sum_k \lambda_{j+1,k}\,\varphi_{j+1,k}(x),$$

then we can also decompose it into its low pass and high pass components:

$$f(x) = \sum_l \lambda_{j,l}\,\varphi_{j,l}(x) + \sum_l \gamma_{j,l}\,\psi_{j,l}(x).$$

The wanted coefficients $\lambda_{j,l}$ and $\gamma_{j,l}$ can be found by calculating the inner products with the dual basis functions:

$$\begin{aligned}\lambda_{j,l} &= \langle f, \tilde{\varphi}_{j,l}\rangle,\\ \gamma_{j,l} &= \langle f, \tilde{\psi}_{j,l}\rangle,\end{aligned}$$

with

$$\langle f,g\rangle \equiv \int_{-\infty}^{\infty} f(x)\,g(x)\,dx.$$

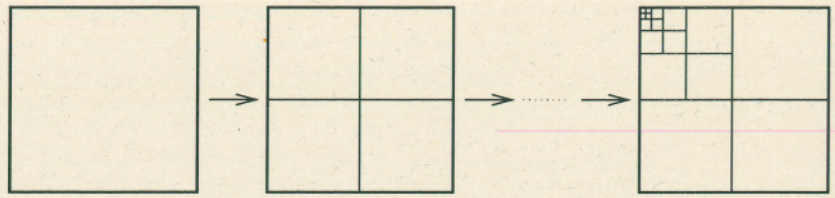


Figure 4: The two-dimensional wavelet transform.

By using these equations we obtain the formulae for the Fast Wavelet Transform:

$$\begin{aligned}\lambda_{j,l} &= \sqrt{2}\sum_k \tilde{h}_{k-2l}\,\lambda_{j+1,k},\\ \gamma_{j,l} &= \sqrt{2}\sum_k \tilde{g}_{k-2l}\,\lambda_{j+1,k}.\end{aligned}$$

2.6 Properties of Wavelets

- Multi-resolution
- Locality in both the time/space domain and the frequency domain
- (bi)Orthogonality
- Smooth basis functions (controlled by vanishing moments)
- Good approximations (in $\mathbf{L}^2$)
- Fast transform algorithms: $\mathcal{O}(n)$
- Stable decompositions

2.7 Higher Dimensions

One step of the wavelet transform of a signal with a dimension n higher than 1 is performed by transforming each dimension of the signal independently. Afterwards the n-dimensional subband that contains the low pass part in all dimensions is transformed further. The two-dimensional case is shown in fig. 4.

2.8 A Simple Compression Example

If we transform an image containing N pixels using a wavelet transform, we have N wavelet coefficients. Now we can consider a very simple image compression algorithm by replacing the M smallest (in absolute value) wavelet coefficients by zero and comparing the result after reconstruction with the original image. A graph showing the Peak Signal to Noise

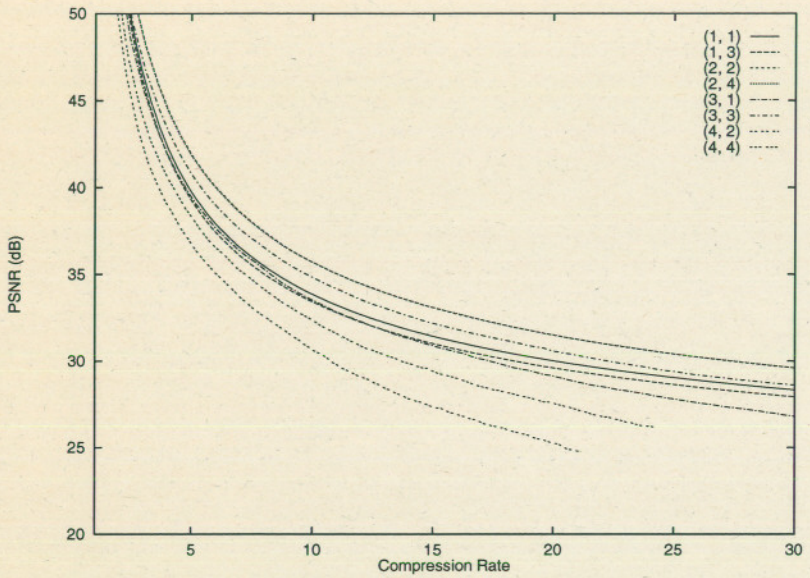


Figure 5: Peak Signal to Noise Ratio (PSNR) versus compression rate for some of the Cohen-Daubechies-Feauveau wavelet transforms.

Ratio versus the compression rate ($\frac{N}{N-M}$) for the well-known "Lena" image and some of the Cohen-Daubechies-Feauveau wavelet transforms is shown in fig. 5. Different test images yield different graphs, but the qualitative behavior of the graphs stay the same.

3 The Lifting Scheme

3.1 Introduction

The Lifting Scheme found its roots in a method to improve a given wavelet transform to obtain some specific properties. Later it was extended to a generic method to create so-called 'Second Generation' wavelets [9, 7, 6, 8].

The generic scheme is represented in fig. 6. Here the filters $\tilde{g}_0$, $\tilde{h}_0$ and the successive subsampling are the 'Lazy' wavelet transform. This transform doesn't do anything but split the original signal S into two sequences containing the odd and the even samples. Afterwards the transform is 'lifted' to a transform with the wanted properties by using one or more of the filter operations s and/or t. Finally the results are normalized by multiplying with the factors n_H respectively n_L.

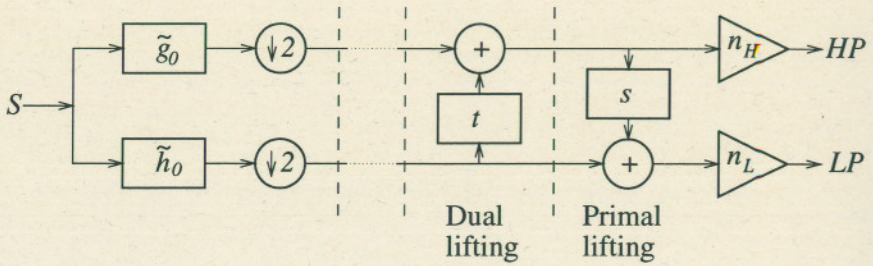


Figure 6: The Lifting Scheme.

The inverse transform is immediately clear from looking at the figure: all we have to do is invert all steps. This is trivial here, unlike with the classical wavelet transform scheme.

Thus the goal is to find one or more lifting filters s and/or t, and normalization factors n_H and n_L, such that this scheme is equivalent to the one in fig. 1. One has proved that all classical wavelet transforms can be implemented using the Lifting Scheme [3].

Advantages of the Lifting Scheme over the classical wavelet transform are:

- Generic method
- Easier to understand (not introduced using the Fourier transform)
- Easier to implement
- Faster ($\times 2$, but still $\mathcal{O}(n)$)
- The inverse transform is easier to find
- The inverse transform has exactly the same complexity as the forward transform.
- Transforms signals with an arbitrary length (need not be 2^n)
- In-place: requires less memory
- Transforms signals with a finite length (without extension of the signal)
- Can be used on arbitrary geometries
- Can be used on irregular samplings
- Can be extended for weighting functions
- All wavelet filters can be implemented using the Lifting Scheme
- Simple extension to an integer transform possible

The result of a wavelet transform using the Lifting Scheme contains all subband data in an interleaved form ('in-place' transform). However, for some applications the classical order ('Mallat' order [5], cfr. fig. 4) with separated subbands is better suited.

3.2 Algorithm

The flow of a wavelet transform algorithm using the Lifting Scheme looks like:

1. Split:

$$\begin{aligned} s_i &\leftarrow x_{2i} \\ d_i &\leftarrow x_{2i+1} \end{aligned}$$

2. Lifting steps: One or more steps k of the form

a. Primal lifting:

$$s_i \leftarrow s_i - \sum_j l_j^k d_j$$

or

b. Dual lifting:

$$d_i \leftarrow d_i - \sum_j l_j^k s_j$$

3. Normalization:

$$\begin{aligned} s_i &\leftarrow n_L \cdot s_i \\ d_i &\leftarrow n_H \cdot d_i \end{aligned}$$

If the transform is normalized, we have $n_L \cdot n_H = 1$.

As you can see, the inverse transform easily follows from this algorithm: just reverse all steps.

3.3 Example: Biorthogonal Cohen-Daubechies-Feauveau

The analyzing filters for some of the popular biorthogonal Cohen-Daubechies-Feauveau wavelets are shown below, together with the lifting steps we obtained by using the technique described in [3]. Note that the coefficients of the filters are always of the form $\frac{z}{2^n}$, with $z \in \mathbf{Z}$ and $n \in \mathbf{N}$. This means that all divisions can be implemented using binary shifts. Unfortunately the coefficients of the lifting steps aren't always of this form.

3.3.1 CDF $(1, x)$

$$\begin{aligned} \tilde{g}_{(1,x)} &: \frac{\sqrt{2}}{2} \cdot \left(1, -1 \right) \\ \tilde{h}_{(1,1)} &: \frac{\sqrt{2}}{2} \cdot \left(1, 1 \right) \\ \tilde{h}_{(1,3)} &: \frac{\sqrt{2}}{16} \cdot \left(-1, 1, 8, 8, 1, -1 \right) \end{aligned}$$

The lifting steps are:

$$\begin{aligned} d_i &\leftarrow d_i - s_i \\ s_i &\overset{(1,1)}{\leftarrow} s_i + \frac{1}{2} d_i \\ s_i &\overset{(1,3)}{\leftarrow} s_i - \frac{1}{16} \left(-d_{i-1} - 8d_i + d_{i+1} \right) \end{aligned}$$

The normalization factors are:

$$\begin{aligned} n_L &= \sqrt{2}, \\ n_H &= \frac{\sqrt{2}}{2}. \end{aligned}$$

3.3.2 CDF $(2, x)$

$$\begin{aligned} \tilde{g}_{(2,x)} &: \frac{\sqrt{2}}{4} \cdot \left(1, -2, 1 \right) \\ \tilde{h}_{(2,2)} &: \frac{\sqrt{2}}{8} \cdot \left(-1, 2, 6, 2, -1 \right) \\ \tilde{h}_{(2,4)} &: \frac{\sqrt{2}}{128} \cdot \left(3, -6, -16, 38, 90, 38, -16, -6, 3 \right) \end{aligned}$$

The lifting steps are:

$$\begin{aligned} d_i &\leftarrow d_i - \frac{1}{2} \left(s_i + s_{i+1} \right) \\ s_i &\overset{(2,2)}{\leftarrow} s_i - \frac{1}{4} \left(-d_{i-1} - d_i \right) \\ s_i &\overset{(2,4)}{\leftarrow} s_i - \frac{1}{64} \left(3d_{i-2} - 19d_{i-1} - 19d_i + 3d_{i+1} \right) \end{aligned}$$

The normalization factors are:

$$\begin{aligned} n_L &= \sqrt{2}, \\ n_H &= \frac{\sqrt{2}}{2}. \end{aligned}$$

3.3.3 CDF $(4, x)$

$$\begin{aligned} \widetilde{g}_{(4,x)} &: \frac{\sqrt{2}}{16} \cdot \left(-1, 4, -6, 4, -1 \right) \\ \widetilde{h}_{(4,2)} &: \frac{\sqrt{2}}{32} \cdot \left(3, -12, 5, 40, 5, -12, 3 \right) \\ \widetilde{h}_{(4,4)} &: \frac{\sqrt{2}}{512} \cdot \left(-10, 40, -2, -192, 140, 560, 140, -192, -2, 40, -10 \right) \end{aligned}$$

The lifting steps are:

$$\begin{aligned} s_i &\leftarrow s_i - \frac{1}{4}\left(d_{i-1} + d_i\right) \\ d_i &\leftarrow d_i - \left(s_i + s_{i+1}\right) \\ s_i &\overset{(4,2)}{\leftarrow} s_i - \frac{1}{16}\left(-3d_{i-1} - 3d_i\right) \\ s_i &\overset{(4,4)}{\leftarrow} s_i - \frac{1}{128}\left(5d_{i-2} - 29d_{i-1} - 29d_i + 5d_{i+1}\right) \end{aligned}$$

The normalization factors are:

$$\begin{aligned} n_L &= 2\sqrt{2}, \\ n_H &= \frac{\sqrt{2}}{4}. \end{aligned}$$

The description of some more lifting steps for CDF wavelets can be found in [10].

4 Multimedia Applications

Due to their locality in space and frequency, and their multi-resolution nature, wavelets have good decorrelating properties and thus they are well suited for signal and image compression and processing (noise removal, contrast enhancement, edge detection, ...). These techniques are very valuable in today's many multimedia (video, audio) applications.

One of the most common applications is image compression. The generic scheme for a wavelet-based (lossy) image compression algorithm is shown in figure 7. The compression consists of a wavelet transform, followed by quantization and encoding steps.

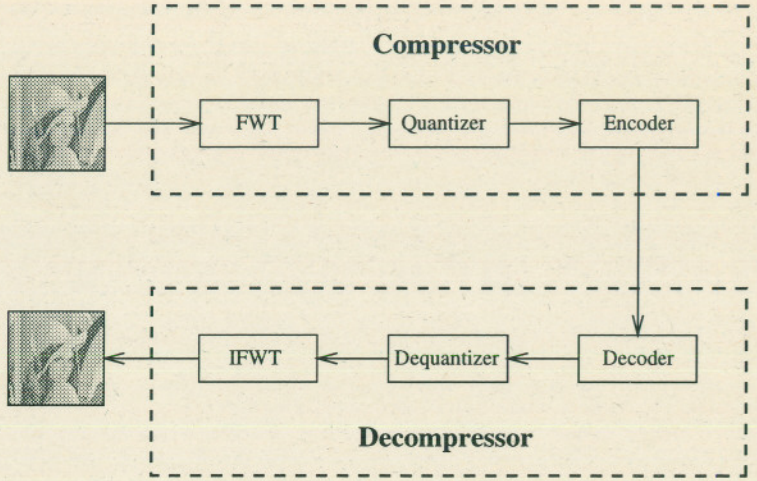


Figure 7: Image compression using wavelets.

In many cases the input data (e.g. a digital image) consist of integer samples only. Unfortunately all of the above transforms assume the input samples are floating point values. They return floating point values as wavelet coefficients, even if the input values actually were integer. It can be considered a waste of memory and CPU resources to first convert integer data to more complex and memory hogging floating point values, especially when we aim at compression. Simply rounding the floating point values to integer values doesn't help because then we will loose the perfect reconstruction feature.

5 The Integer Wavelet Transform

The first steps in the direction of integer wavelet transforms where set with the *S* and *TS* transform in *CREW* (Compression with Reversible Embedded Wavelets) [11, 12].

Recent research showed that the lifting scheme can be easily modified to a transform that maps integers to integers and that is reversible, and thus allows a perfect reconstruction [1]. This will be done by adding some rounding operations, at the expense of introducing a non-linearity in the transform.

A lifting step basically looks like

$$x_i \leftarrow x_i - \frac{1}{a}\sum_j b_j y_j.$$

Here we assume that $a, b_j \in \mathbf{Z}$, i.e. they are integers. This is true for various wavelet transforms, among which the Cohen-Daubechies-Feauveau biorthogonal wavelets.

This lifting step can be modified in one of the following ways. A truncation of x to the largest integer smaller than x will be indicated by $\lfloor x \rfloor$.

Full rounding The result of the division by a is rounded:

$$\tilde{x}_i \leftarrow x_i - \left\lfloor \frac{\sum_j b_j y_j + \frac{a}{2}}{a} \right\rfloor .$$

Here $\tilde{x}_i$ is an integer approximation to x_i.

Without rounding We avoid the division by a by multiplying the other terms with a:

$$\tilde{x}_i \leftarrow a x_i - \sum_j b_j y_j .$$

Here $\tilde{x}_i = ax$, and thus the dynamic range of the wavelet coefficients will increase. This has to be taken into account in later steps. Note that in this case no real rounding is performed, and thus this can be considered to be an 'exact' implementation of the floating point version, yielding integers.

Mixed form We combine the rounding and multiplication steps of both methods:

$$\tilde{x}_i \leftarrow a_1 x_i - \left\lfloor \frac{\sum_j b_j y_j + \frac{a_2}{2}}{a_2} \right\rfloor ,$$

with

$$\begin{aligned} a_1, a_2 &\in \mathbf{Z}, \\ a_1 \cdot a_2 &= a. \end{aligned}$$

This variant can be used if we want to have more control over the dynamic range of the resulting $\tilde{x}_i$.

Note that in all of the three cases above the modified lifting step is still reversible, and thus the perfect reconstruction feature is still present.

If $a, b_j \notin \mathbf{Z}$, we can still use the 'full rounding' method to obtain a transform that maps integers to integers.

Examples

CDF $(1, x)$ This is the well-known Haar transform. The lifting steps

$$\begin{aligned} d_i &\leftarrow d_i - s_i, \\ s_i &\leftarrow s_i + \frac{1}{2} d_i, \end{aligned}$$

can be changed into

$$\begin{aligned} d_i &\leftarrow d_i - s_i, \\ s_i &\leftarrow s_i + \left\lfloor \frac{1}{2} d_i \right\rfloor , \end{aligned}$$

yielding a transform that maps integers to integers, if we forget about the normalization factors.

CDF $(2, x)$ The lifting steps

$$\begin{aligned} d_i &\leftarrow d_i - \frac{1}{2}(s_i + s_{i+1}), \\ s_i &\leftarrow s_i - \frac{1}{4}(-d_{i-1} - d_i), \end{aligned}$$

can be changed into

$$\begin{aligned} d_i &\leftarrow 2d_i - s_i - s_{i+1}, \\ s_i &\leftarrow s_i + \left\lfloor \frac{d_{i-1} + d_i + 4}{8} \right\rfloor . \end{aligned}$$

Note that the divider in the second lifting step is changed from 4 to 8 to compensate for the multiplication with 2 in the first step. An alternative, with rounding in both steps, would be:

$$\begin{aligned} d_i &\leftarrow d_i - \left\lfloor \frac{s_i + s_{i+1}}{2} \right\rfloor , \\ s_i &\leftarrow s_i + \left\lfloor \frac{d_{i-1} + d_i + 2}{4} \right\rfloor . \end{aligned}$$

More information about the implementation aspects can be found in [10].

References

[1] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo. Wavelet transforms that map integers to integers. Technical report, Department of Mathematics, Princeton University, 1996.

[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 and W. Sweldens. Factoring wavelet transforms into lifting steps. Technical report, Bell Laboratories, Lucent Technologies, 1996.

[4] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36(3):377–412, 1994.

[5] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Anal. Mach. Intell.*, 11(7):674–693, 1989.

[6] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *Computer Graphics, (SIGGRAPH '95 Proceedings)*, 1995.

[7] 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.

[8] W. Sweldens. The lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A. F. Laine and M. Unser, editors, *Wavelet Applications in Signal and Image Processing III*, pages 68–79. Proc. SPIE 2569, 1995.

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

[10] G. Uytterhoeven, D. Roose, and A. Bultheel. Wavelet transforms using the Lifting Scheme. ITA-Wavelets Report WP 1.1, Department of Computer Science, Katholieke Universiteit Leuven, Belgium, November 1996.

[11] A. Zandi, M. Boliek, E. L. Schwartz, M. J. Gormish, and J. D. Allen. *CREW Lossless/Lossy Image Compression — Contribution to ISO/IEC JTC 1.29.12.* RICOH California Research Center, June 1995.

[12] A. Zandi, M. Boliek, E. L. Schwartz, and A. Keith. *CREW Second Evaluation - ISO/IEC JTC1/SC29/WG1.* RICOH California Research Center, November 1995.