CONVOLUTION USING THE UNDECIMATED DISCRETE WAVELET TRANSFORM

Haitao Guo *C.S. Burrus*

Department of Electrical and Computer Engineering - MS366
Rice University, Houston, TX 77251-1892
harry@rice.edu *csb@rice.edu*

ABSTRACT

Convolution is one of the most widely used digital signal processing operations. It can be implemented using the fast Fourier transform (FFT), with a computational complexity of $O(N \log N)$. The undecimated discrete wavelet transform (UDWT) is linear and shift invariant, so it can also be used to implement convolution. In this paper, we propose a scheme to implement the convolution using the UDWT, and study its advantages and limitations.

1. INTRODUCTION

Convolution is the fundamental operation of linear system theory, and discrete convolution is one of the most widely used digital signal processing operation. Finite impulse response (FIR) digital filters are designed to be convolved with input signals to achieve certain effects, and the fast Fourier transform (FFT) is mostly used to implement convolution. Therefore any scheme that can speed up the convolution process is theoretically interesting and practically important.

The Fourier transform, Laplace transform and Z-transform all have similar convolution theorems, which relate a signal domain convolution to a transform domain product. The wavelet transform is a powerful new mathematical tool. It would desirable to have a similar convolution theorem for the wavelet transform. It has been shown, however that the continuous wavelet transform can not admit a Fourier type convolution theorem [1], but we can convolve two signals by directly *convolving* the sub-band signals and combining the results [2, 3]. Neither of the above answers are quite satisfactory. The lack of shift invariance is one of the reasons for the non-existence of a wavelet convolution theorem of the Fourier type.

The undecimated discrete wavelet transform (UDWT) is linear and shift invariant, so it could be used to implement convolution. The computational complexity of the UDWT is $O(N \log N)$, which is of the same order of the FFT. We propose a scheme to implement convolution using the UDWT, and study its advantages and limitations.

In this paper, we only consider discrete circular convolution, since it is easy to convert a linear convolution into a circular convolution. Section 2 contains a brief discussion of the UDWT and a detailed analysis of its computational complexity. The main part of this paper is in section 3, where we study the proposed scheme to implement convolution using the UDWT. Some examples are given in section 4.

The research is supported by BNR and Texas-ATP grant.

2. UNDECIMATED DISCRETE WAVELET TRANSFORM

2.1. A Review

The undecimated discrete wavelet transform has been independently discovered several times, for different purposes and under different names [4, 5, 6, 7, 8, 9], e.g. the shift/translation invariant wavelet transform, the stationary wavelet transform, or the redundant wavelet transform. The key point is that it is redundant, shift invariant, linear, and it gives a better approximation to the continuous wavelet transform than the approximation provided by the orthonormal (ON) discrete wavelet transform (DWT). A discussion of the algorithm and history can be found in [7]. Here we only show the UDWT from the matrix point of view. The undecimated discrete wavelet transform can be visualized as a matrix multiplication

$$\mathbf{Y} = \mathbf{W}\mathbf{y}, \tag{1}$$

where $\mathbf{y}$ is a $1 \times N$ input vector, $\mathbf{W}$ is an $(L+1)N \times N$ matrix, where L is the number of levels or scales of decomposition, and $\mathbf{Y}$ is the $(L+1)N \times 1$ output vector. Let's denote

$$\mathbf{W} = [\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_L, \mathbf{W}_{L+1}]^T, \tag{2}$$

where $\mathbf{W}_i$ is an $N \times N$ matrix, and the columns of $\mathbf{W}_i$ are circularly shifted versions of a single vector $\mathbf{w}_i$, which is the usual discrete wavelet transform (DWT) basis at i_{th} scale ($i = 1$ for the finest scale), and $\mathbf{w}_{L+1}$ is the scaling function at the coarsest scale.

Because the system is under determined, there are many inverse transforms, but a somewhat desirable one is given by

$$\mathbf{M} = \left[\frac{1}{2}\mathbf{W}_1, \frac{1}{2^2}\mathbf{W}_2, \ldots, \frac{1}{2^L}\mathbf{W}_L, \frac{1}{2^L}\mathbf{W}_{L+1}\right], \tag{3}$$

where the scaling factors $(\frac{1}{2}, \frac{1}{4}, \cdots \frac{1}{2^L}, \frac{1}{2^L})$ are required to offset the increasing redundancy of the UDWT as the scale becomes coarser.

Direct multiplication in (1) requires $O((L+1)N^2)$ operations. Fortunately a fast algorithm exists [4], so that the total number of operations is $O(LN)$, which is at most $O(N \log N)$.

2.2. Complexity of the Wavelet Transforms

In order to establish the efficiency of the UDWT based convolution algorithm, we need to study the computational complexity of the wavelet transforms. Let us assume the length of the input signal is N, the length of the conjugate quadrature filter (CQF) is M, and the number of levels of decomposition is L. The basic step of DWT is the convolution of the input signal with the CQF, and the efficient implementation has the lattice structure [10]. The number of multiplication and additions needed to convolve the signal with both the high pass and the low pass CQF are[1]

$$\mathcal{M}_{CQF}(N,M) = MN, \tag{4}$$

$$\mathcal{A}_{CQF}(N,M) = MN. \tag{5}$$

Throughout this paper, we use $\mathcal{M}$ to denote the number of real multiplies, and $\mathcal{A}$ for the number of real additions, various subscripts are used for different algorithms. Due to the lattice structure, the complexities in (4) and (5) are nearly half of what is normally required for straight forward convolution. If down-sampling is performed after the convolution, the above complexities are further cut by half.

For the orthonormal (ON) DWT,

$$\mathcal{M}_{DWT}(N,M,L) = MN, \tag{6}$$

$$\mathcal{A}_{DWT}(N,M,L) = MN. \tag{7}$$

They are independent of the number of levels of decomposition, since successive down-samplings are carried out at each level or scale.

For an L level UDWT, since we do not down-sample, the total complexity increases linearly with L, as

$$\mathcal{M}_{UDWT}(N,M,L) = MNL, \tag{8}$$

$$\mathcal{A}_{UDWT}(N,M,L) = MNL. \tag{9}$$

The number of levels L is bounded above by $\log_2 N$, so

$$\mathcal{M}_{UDWT}(N,M,L) \le MN\log_2 N, \tag{10}$$

$$\mathcal{A}_{UDWT}(N,M,L) \le MN\log_2 N. \tag{11}$$

We need to point out that DWT and UDWT are themselves sequences of convolutions. Depending on the practical situation, we might implement those convolutions using other fast algorithms like the FFT [11], e.g. for cosine modulated orthogonal wavelet transform [12].

3. CONVOLUTION USING UDWT

3.1. The Scheme

Let $\mathbf{d}$ be an $1 \times (L+1)$ vector, and $\mathbf{o}$ be an $1 \times N$ vector of ones, then the Kronecker tensor product $\mathbf{k} = \mathbf{d} \otimes \mathbf{o}$ is a vector of length $(L+1)N$. Let $\mathbf{D}$ be a diagonal matrix with the elements of $\mathbf{k}$ on the diagonal, then $\mathbf{MDW}$ is a linear shift[2] invariant operator, i.e.

$$\mathbf{MDW}(a\mathbf{x} + b\mathbf{y}) = a\mathbf{MDWx} + b\mathbf{MDWy}, \tag{12}$$

$$\mathbf{MDW}S(\mathbf{x}, I) = S(\mathbf{MDWx}, I), \tag{13}$$

where $S(x, I)$ denotes the circular shift of $\mathbf{x}$ by I units. So $\mathbf{MDW}$ performs a circular convolution, as

$$\mathbf{x} * \mathbf{h} = \mathbf{MDWx}, \tag{14}$$

where $\mathbf{h}$ is determined by $\mathbf{d}$, and the CQF. Thus we have a scheme that implements convolution using UDWT. We can further merge those scaling factors $(\frac{1}{2}, \frac{1}{4}, \cdots \frac{1}{2^L}, \frac{1}{2^L})$ in $\mathbf{M}$ into the diagonal matrix $\mathbf{D}$, and simplify our method as $\mathbf{W}^T\hat{\mathbf{D}}\mathbf{W}$.

Our UDWT based convolution method has a form similar to the convolution theorem of the Fourier transform, i.e. transform domain product. However, that is only true for one of the input signals ($\mathbf{x}$), the other input $\mathbf{h}$ is buried in the CQFs and $\mathbf{d}$. Or we can equivalently say that $\mathbf{h}$ is transformed to $\mathbf{d}$ through some other transformation involving the CQF. This is not surprising, since the wavelet transform that has the exact Fourier type convolution theorem does not exist [1]. Another drawback is, as we will discuss in section 3.2, only certain types of convolutions can be implemented using UDWT. Note that $\mathbf{W}^T\hat{\mathbf{D}}\mathbf{W}$ can be interpreted as a scale weighting operation.

3.2. Limitations

To characterize those convolutions that can be implemented using UDWT, we need to study our scheme in greater detail. Let $\mathbf{h}$ denote the convolution kernel for $\mathbf{W}^T\hat{\mathbf{D}}\mathbf{W}$, i.e. $\mathbf{x}*\mathbf{h} = \mathbf{W}^T\hat{\mathbf{D}}\mathbf{W}\mathbf{x}$. We can show that

$$\mathbf{h} = \sum_{i=1}^{L+1} d_i \mathbf{a}_i, \tag{15}$$

where d_i is the scaled effective weighting factor for the i_{th} scale, and $\mathbf{a}_i$ is the autocorrelation (AC) sequence of $\mathbf{w}_i$. So the filter that we actually use is a weighted combination of the autocorrelation sequences of the wavelets on different scales. Since the autocorrelation sequences are symmetric and of odd length[3], $\mathbf{h}$ must also be symmetric and of odd length, i.e. the type-I filter in [13]. So the degrees of freedom of our scheme are upper bounded by $N/2+1$.

It is thus important to ask whether all the odd length symmetric filters can be implemented using UDWT, i.e. whether the bound $N/2+1$ is tight. For a given $\mathbf{h}$, we need to solve a set of *nonlinear* equations to get *both* the CQFs and $\mathbf{d}$. These equations are not straight forward to solve, so one might want to design filters that has the structure as in (15). Since all the CQF can be parameterized [10], the design task is an unconstrained nonlinear optimization problem, for which general tools exist [14].

3.3. Complexity of Convolution Algorithms

Because of the structure, the multiplication of $\mathbf{D}$ can be merged into the UDWT, and does not require any additional computations. So the total number of multiplications and additions needed to perform our circular convolution scheme is

$$\mathcal{M}_{UDWTConv}(N,M,L) = 2MNL, \tag{16}$$

and

$$\mathcal{A}_{UDWTConv}(N,M,L) = 2MNL. \tag{17}$$

The straight forward convolution requires

$$\mathcal{M}_{Conv}(N, \mathcal{L}(\mathbf{h})) = N\mathcal{L}(\mathbf{h}), \tag{18}$$

and

$$\mathcal{A}_{Conv}(N, \mathcal{L}(\mathbf{h})) = N\mathcal{L}(\mathbf{h}), \tag{19}$$

operations, where $\mathcal{L}(\mathbf{h})$ is the length of $\mathbf{h}$. For FFT based convolution,

$$\mathcal{M}_{FFTConv}(N) = N\log_2 N + N, \tag{20}$$

and

$$\mathcal{A}_{FFTConv}(N) = \frac{7}{2}N\log_2 N. \tag{21}$$

In order for our algorithm to be the most efficient, i.e. having the least total number of multiplications and additions, we need

$$\mathcal{L}(\mathbf{h}) > 2ML, \tag{22}$$

[1] First order approximation.

[2] We only consider the circular shifts in this paper.

[3] Ignoring the circular nature of our autocorrelation at this moment.

and

$$N > 2^{\frac{8}{9}ML}. \tag{23}$$

For sufficiently long $\mathbf{x}$, (23) is easily satisfied. We will show in section 3.4 that we can generate rather long filter by short CQF and few levels of decomposition, such that (22) is valid. For these cases, we indeed have a fast algorithm to implement the convolution.

3.4. Size of the AC sequences of DWT basis

Let $\mathcal{L}(\mathbf{s})$ denote the length/support of the vector $\mathbf{s}$. Since the DWT basis sequences at different levels are related by up-sampling and convolution, it can be shown that[4]

$$\mathcal{L}(\mathbf{w}_i) = 2\mathcal{L}(\mathbf{w}_{i-1}) + M - 2, \qquad \mathcal{L}(\mathbf{w}_1) = M. \tag{24}$$

The length of the autocorrelation function and the length of the DWT basis are related by

$$\mathcal{L}(\mathbf{a}_i) = 2\mathcal{L}(\mathbf{w}_i) - 1. \tag{25}$$

Solving the difference equation (24), we have

$$\mathcal{L}(\mathbf{w}_L) = 2^L(M-1) - M + 2, \tag{26}$$

and

$$\mathcal{L}(\mathbf{a}_L) = 2^{L+1}(M-1) - 2\,M + 3. \tag{27}$$

So $\mathcal{L}(\mathbf{a}_L)$ grows exponentially with L [5], thus $\mathcal{L}(\mathbf{h})$ also grows exponentially with L, and the growth rate is independent of M. We can see that (22), which only requires a linear growth rate with L, can be easily satisfied. If we measure the efficiency of our algorithm by the ratios between the computational complexity of other algorithms and the complexity of our UDWT based algorithm, we can conclude that the efficiency grows logarithmically with N and exponentially with L. We need to emphasis that the filters that we can efficiently implement must have the structure as (15). There might exist other efficient algorithms to implement such filters, however we are not aware of any such algorithm.

4. EXAMPLE AND DISCUSSIONS

As an example, we would like to design a lowpass FIR filter with a cutoff frequency at 0.125π. For simplicity, we only minimize the l_2 difference between the ideal filter and the designed filter. We use three level UDWT implementation, and length 6 CQFs on each level. In order to gain more freedom, we use different CQFs on different levels. However, this does not change the total complexity of the implementation. We need to find three lattice parameters on each level, and four weighting coefficients. The unconstrained optimization routine in [14] were used to find the 14 unknowns that minimize the l_2 difference between the ideal filter and the designed filter, and the resulting[6] CQF coefficients[7] and weighting vector are shown in Tab. 1 and 2 respectively. The frequency response of the final filter is shown in Fig. 1(a). The resulting filter has the same l_2 error as a length-53 optimal FIR filter, whose frequency response is shown in Fig. 1(b). Comparing Fig. 1(a) and Fig. 1(b), we can see that the new filter has a slight higher overshoot, but its frequency response in stopband is much better. The

[4] Ignore the warping caused by the circular convolution.

[5] We assume N is very large, since $\mathcal{L}(\mathbf{a}_L) < N$.

[6] One local minimum.

[7] The constrain on the sum of squares of the CQF coefficients is relaxed, and it does not affect the problem.

Table 1: CQF coefficients for the example lowpass filter.

L	1	2	3
	0.618744	0.527011	0.012580
	1.023193	0.760529	0.537700
	-0.310055	0.341187	0.803249
	0.135727	-0.193336	0.213485
	-0.135082	-0.068969	-0.156461
	0.081686	0.047792	0.003661

Table 2: Weighting coefficients for the autocorrelation sequences.

i	1	2	3	4
d_i	0.201496	-0.121795	-0.001087	-0.000692

computational complexity of the new filter is the same as a length-36 FIR filter. For comparison, we also plot the frequency response of a length-36 optimal FIR lowpass filter in Fig. 1(c). Clearly, for this example, we can efficiently implement better filter using the UDWT.

The frequency responses of the four autocorrelation sequences are shown in Fig. 1(d). Since the Fourier transform is linear, the frequency response in Fig. 1(a) is the weighted combination of the frequency responses of the autocorrelation sequences, and the weights are the same as in Tab. 2.

During our limited experiments, we observe that the UDWT implementations is very efficient when the lowpass bandwidth is $\frac{\pi}{2^N}$, and the efficiency grows when the passband gets narrower. This may due to the tree structure of the wavelet transform, where we successively split the lowpass band. For other kind of cutoff frequencies or filter types, undecimated wavelet packet transform or M-band undecimated wavelet transform might be helpful. Multirate implementations of narrow-band lowpass filters have been studied in [15], however the implementations there are not strictly shift invariant.

Up-to now, we only consider traditional software implementation of the wavelet transform. Recent result [16] shows that the CORDIC-based hardware implementation of the wavelet transform has very nice properties, e.g. it can be efficiently approximated in reduced parameter space and implemented with a small number of shift & add operations. So we can extend our undecimated wavelet transform based convolution algorithm for the CORDIC-based hardware implementation. We thus need to optimize in the reduced parameter space, so the overall system has efficient hardware implementation and desired filter performance.

5. SUMMARY

We propose a novel method to implement the convolution using the undecimated wavelet transform. Similar to the convolution theorem of the Fourier transform, the UDWT based convolution has the form of a transform domain product. The filters that can be implemented using the UDWT are completely characterized. We also show that for certain cases, our method is much more efficient than the traditional convolution algorithms.

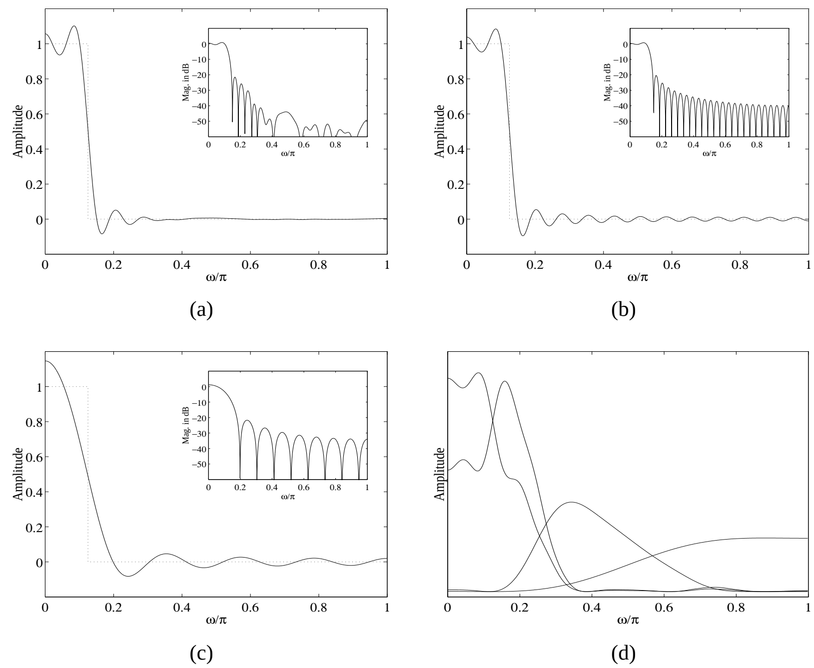

Figure 1: Frequency responses of lowpass filters with cutoff frequency at 0.125π. (a) The designed filter that can be efficient implemented using the UDWT. (b) The length-53 optimal filter that has the same l_2 error as the filter in (a). (c) The length-36 optimal filter that has the same computational complexity as the filter in (a). (d) The frequency responses of the four autocorrelation sequences.

REFERENCES

[1] A. R. Lindsey. The non-existence of a wavelet function admitting a wavelet transform convolution theorem of the Fourier type. Technical report, Ohio University, Athens, Ohio, August 1994.

[2] P. P. Vaidyanathan. Orthonormal and biorthonormal filter-banks as convolvers, and convolutional coding gain. *IEEE Trans. SP*, June 1993.

[3] S. M. Phoong and P. P. Vaidyanathan. The biorthonormal filter bank convolver, and applications in low sensitivity FIR filter structures. *IEEE Proc. Int. Conf. Acoust., Speech, Signal Processing*, III:165–168, 1993.

[4] G. Beylkin. On the representation of operators in bases of compactly supported wavelets. *SIAM J. Numer. Anal.*, 29(6):1716–1740, 1992.

[5] M. J. Shensa. The discrete wavelet transform: Wedding the à trous and Mallat algorithms. *IEEE Trans. Inform. Theory*, 40:2464–2482, 1992.

[6] S. Mallat. Zero-crossings of a wavelet transform. *IEEE Trans. Inform. Theory*, 37(4), July 1991.

[7] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells, Jr. Nonlinear processing of a shift-invariant DWT for noise reduction. In *SPIE conference on wavelet applications*, volume 2491, Orlando, FL, April 1995. Also Tech. report CML TR95-03, Rice University, Houston, TX.

[8] G. P. Nason and B. W. Silverman. Stationary wavelet transform and some statistical applications. Technical report, Department of Mathematics, University of Bristol, Bristol, U.K., February 1995. Also Tech. Report.

[9] H. Guo. Theory and applications of the shift-invariant, time-varying and undecimated wavelet transform. Master's thesis, Rice University, Houston, TX, May 1995.

[10] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, Englewood Cliffs, NJ, 1992.

[11] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall, Englewood Cliffs, NJ, 1995.

[12] R. A. Gopinath and C. S. Burrus. On cosine-modulated wavelet orthonormal bases. *IEEE Trans. on Image Processing*, 4(2):162–176, February 1995. Also Tech. Report No. CML TR92-06, 1992.

[13] T. W. Parks and C. S. Burrus. *Digital Filter Design*. John Wiley & Sons. Inc., New York, NY, 1st edition, 1987.

[14] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1st edition, 1983.

[15] R. E. Crochiere and L. R. Rabiner. *Multirate Digital Signal Processing*. Prentice-Hall, Englewood Cliff, NJ, 1983.

[16] P. Rieder, K. Gerganoff, J. Götze, and J. A. Nossek. Parameterization and implementation of orthonormal wavelet transforms. Technical report, Technical University of Munich, Network Theory & Circuits Design, August 1995. Submitted to ICASSP 1996, Atlanta, GA.