Overcomplete Lifted Wavelet Representations for Multiscale Feature Analysis

Minbo Shim

Computer and Information Science and Engineering
University of Florida
Gainesville, FL 32611
ms2@cise.ufl.edu

Andrew Laine

Center for Biomedical Engineering
Columbia University
New York, NY 10027
laine@bme.columbia.edu

Abstract

The lifting scheme was introduced as a flexible tool to construct compactly supported second generation wavelets and wavelet transform. However because it is not translation invariant, the traditional lifting framework may not be good for multiscale feature analysis where a translation-invariant characteristic are highly desirable. In this paper we address the following question: Can the lifting scheme be used as a framework for overcomplete wavelet representations with multiscale feature analysis in mind? We address this question by investigating each stage of the multiscale analysis: Split, Dual lifting and Primal lifting. We introduce a smoothing Lazy wavelet in the split stage. We then show that only the dual lifting is necessary since the primal lifting required to compensate for the aliasing needs no longer exist. We also demonstrate that the proposed scheme achieves better performance without introducing boundary artifacts that exist in the traditional methods.

1. Introduction

The lifting scheme was introduced as a flexible tool for constructing compactly supported second generation wavelets which are not necessarily translates and dilates of one *mother* wavelet [8]. The idea behind the lifting scheme is to use a simple basis function and build a better performing one with desirable properties. Flexibility afforded by the lifting scheme allows basis functions to change their shapes near the boundaries without degrading regularities. The lifting scheme also provides faster processing by making optimal use of similarities between high and low pass filters to speed up the calculation of transform coefficients [7].

The lifting scheme utilizes a classical 2-channel filter bank as a framework for multiresolution analysis. However the traditional framework is not translation invariant since the downsampling and upsampling operations are not translation invariant. Representations with a translation-invariant characteristic are highly desirable for feature analysis. Mallat and Zhong introduced an adaptive sampling based upon local maxima as an overcomplete wavelet representation [4].

In this paper we address the following question: Can the lifting scheme be used as a framework for overcomplete wavelet representations with feature analysis in mind? We investigate each stage of the multiscale framework described in [7]. We use a *smoothing Lazy* wavelet in the *split* stage which does not subsample, but smooths an input image so that the low-pass channel contains redundant information. We then show that the following *dual* lifting stage alone, based upon specific properties, indeed leads to a useful multiresolution framework for feature analysis. The primal lifting required to compensate for the aliasing needs no longer exist in overcomplete representations.

We also investigate how much speedup the proposed algorithm can achieve over the traditional overcomplete wavelet transforms by comparing the computational costs defined by the number of multiplications and additions.

This paper is organized as follows: in Section 2 the overcomplete lifting scheme is presented. Section 3 describes a multiscale lifted block algorithm that speeds up the performance of the convolution. In Section 4 we compute and compare the computational costs of various overcomplete DWT algorithms and show which one is the most efficient.

2. Overcomplete Lifting Scheme

Traditionally multiresolution analysis is implemented through a refinement relation consisting of a scaling function and wavelets with finite filters. The dual functions also generate a multiresolution analysis with dual filters. It has been shown that a set of finite biorthogonal filters

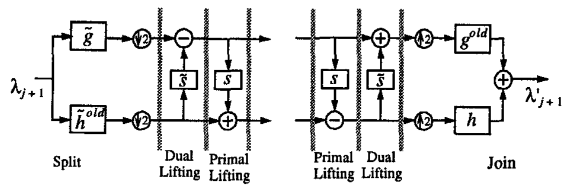


Figure 1. Wavelet transform with lifting schemes.

guarantees the existence of a set of biorthogonal functions [8]. Thus a relation of biorthogonal filters established from the Vetterly-Herley lemma leads to defining a lifting scheme as

$$\psi(x) = \psi^{old}(x) - \sum_k s_k \varphi(x-k)$$

$$\tilde{\varphi}(x) = \tilde{\varphi}^{old}(x) + \sum_k s_{-k} \tilde{\psi}(x-k) \quad (1)$$

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

Through the trigonometric polynomial s, we have a full control over wavelets and dual functions. These lifting schemes are related to interpolating scaling functions which may be generated either by subdivision [2] or average interpolation [3]. By alternating primal and dual lifting, one can come up with a multiresolution analysis geared for specific properties. This alternation leads to a canonical form of wavelet transform shown in Figure 1. Here the filters followed by successive subsampling define the *Lazy* wavelet transform which simply *splits* an original signal into odd and even subsets. The dual lifting *predicts* wavelet coefficients with the help of the scaling function coefficients based upon the correlation present in the finer signal. The primal lifting then *updates* the scaling function coefficients using the previously predicted wavelet coefficients to remove possible aliasing resulted from the downsampling [7].

For overcomplete representations, we use a *smoothing lazy wavelet* in the split stage which does not downsample the input image so that $\lambda_{j+1,k} = \lambda_{j,l} = \gamma_{j,m}$. We then smooth $\lambda_{j,k}$ using a smoothing filter, which is often a Gaussian [4]. This new splitting methodology in the filter bank scheme is then followed by the decorrelation of wavelet coefficients using the dual lifting. Since the new *split* operation does not use downsampling to make distinct subsets $\{\lambda_{j,l}\}$ and $\{\gamma_{j,l}\}$, there is no aliasing. The primal lifting introduced to compensate for the aliasing by preserving energy between two contiguous approximations needs no longer exist. The new forward and inverse wavelet transform algorithm for overcomplete representations may thus be formulated as

<u>Forward Transform</u>

Split: $\lambda_{j,l} = \sqrt{2}\sum_k \tilde{h}_{k-2l}\lambda_{j+1,k}$, $\gamma_{j,m} = \lambda_{j+1,k}$

Dual Lifting: $\gamma_{j,m} = \gamma_{j,m} - \sum_{n=0}^{N-1} \tilde{s}_n \lambda_{j,m-2n}$

<u>Inverse Transform</u>

Dual Lifting: $\gamma_{j,m} = \gamma_{j,m} + \sum_{n=0}^{N-1} \tilde{s}_n \lambda_{j,m-2n}$

join: $\lambda_{j+1,k} = \lambda_{j,l} + \gamma_{j,m}$

where $K(j+1)$, $L(j)$ and $M(j)$ are index sets such that $|K(j+1)| = |M(j)| = K(j)$, $\{k | k \in K(j+1)\}$, $\{l | l \in L(j)\}$ and $\{m | m \in M(j)\}$. $\tilde{N}$ and N are primal and dual vanishing moments, respectively. Figure 2 illustrates how to compute the wavelet coefficients with a cubic dual lifting polynomial (N = 4). The predicted wavelet coefficients contain the amount by which the coarser approximation locally fails to be cubic. Here the dual lifting coefficients $\tilde{s}$ = {-0.0625, 0.5625, 0.5625, -0.0625}. The coefficients are adjusted near the boundaries of the finite input signal to avoid boundary effects [7]. For N = 6, $\tilde{s}$ = {0.0117, -0.0977, 0.5859, 0.5859, -0.0977, 0.0117} and $\tilde{s}$ = {-0.0024, 0.0239, -0.1196, 0.5981, 0.5981, -0.1196, 0.0239, -0.0024} for N = 8.

Figure 3 shows wavelet representations for two levels of analysis. We used a cubic spline-based filter for H described in [4] for the smoothing lazy wavelet transform. For the dual lifting, we applied a cubic polynomial (N = 4) with coefficients $\tilde{s}$ = {-0.0625, 0.5625, 0.5625, -0.0625}.

3. Multiscale Lifted Block Algorithm

More generalized version of multiresolution analysis via the second generation wavelets is shift-variant. In other

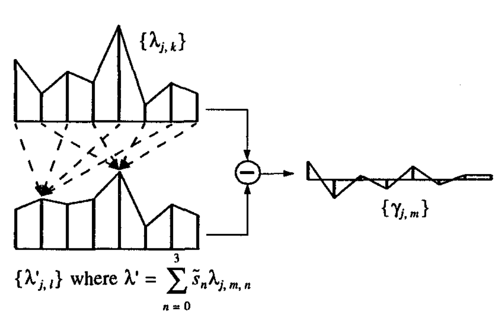


Figure 2. Forward wavelet transform to extract detail information as failures to be cubic.

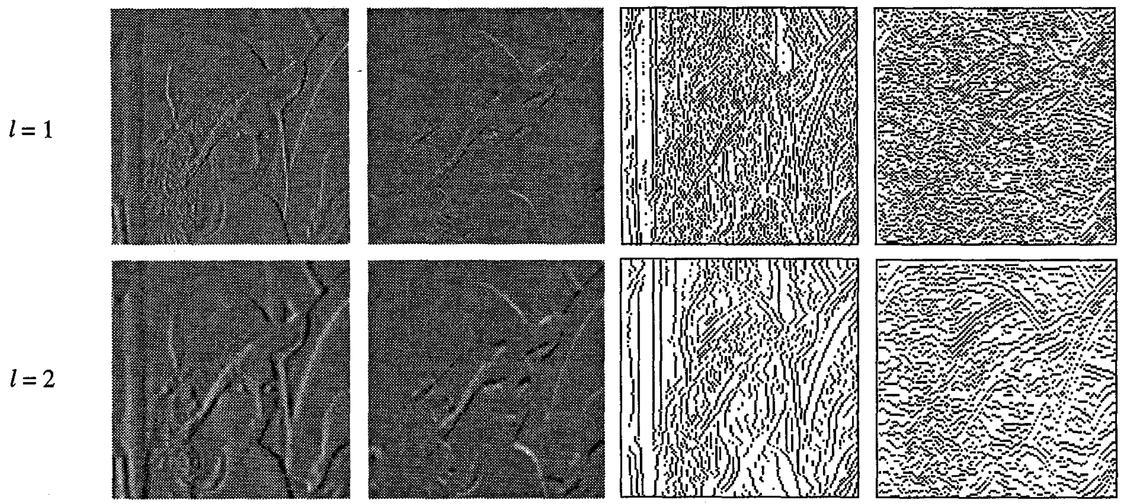


Figure 3. Wavelet transform of Lenna image with the proposed scheme. The first two columns from the left give the wavelet coefficients in horizontal and vertical direction, respectively. The last two columns show the local maxima.

words it accommodates finite data sets by introducing different, specially adapted wavelets near the boundaries. Since the basis functions are not translations and dilations of one specific function any more, the Fourier transform can no longer be used as an implementation tool for the second generation wavelets. Then the wavelet transform must rely on a spatial convolution to carry out multiresolution analysis.

Shim and Laine [6] showed that there exists a way to improve the performance of the traditional convolution by a *multiscale block algorithm* which provides the best performance in computing the discrete wavelet transform for overcomplete representations.

In frame-based multiscale representations, spatial convolution require 2^l-1 zeros between each non-zero filter coefficients in a filter kernel as scale changes. This increases the computational cost exponentially. However, the multiscale block algorithm reduces the cost by a factor of 2^l. The basic idea behind this algorithm is to keep the size of the filter kernels unchanged throughout multiscale analysis. Instead, we partition the input image into four sub-regions so that each region contains only those pixels affected by the filter kernel (without expansion).

Figure 4 illustrates differences between the two approaches: a linear convolution with zero-paddings in the kernel, and an alternative method by splitting the convolved image into four sub-regions for further convolution at finer levels. In the new method shown in Figure 4(b), the input image is first convolved with the original filter kernel at level 1. The convolved image is then divided into four sub-regions such that each region consists of only those pixels processed by the convolution at the next level ($l = 2$). After the splitting, convolutions are then applied to each of the sub-regions independently. Synthesis is simply the reverse of analysis. In other words, we first merge four sub-regions at a coarser level into a larger region, then convolve it with a filter kernel.

For quantitative comparison, let us compute the computational cost of each method of implementation. For simplicity, we focus on the analysis only. Let the size of the input image and the size of the original filter kernel be n and m, respectively. And let us denote the level of analysis by l. Then, in the traditional linear convolution case, the computational cost C_A is

$$C_A = n^2 \cdot (m + 2^l - 2)^2 ,$$

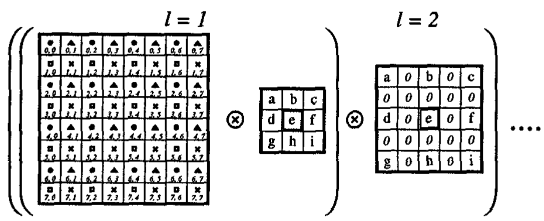
and in our approach, the computational cost C_B is

$$C_B = n^2 \cdot 2^{l-1} \cdot m^2 .$$

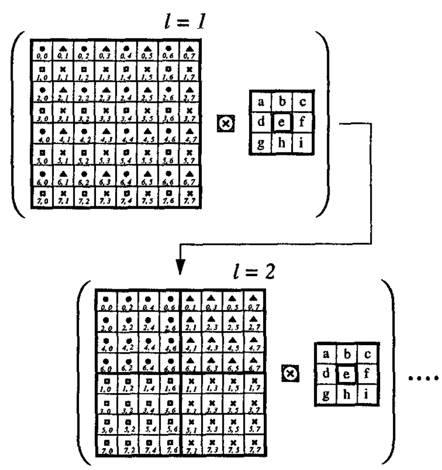
Normally, the size of the filter kernel is small, so we can assume m is a constant. The speedup by the new approach is then

$$O\left(\frac{C_A}{C_B}\right) \approx O\left(\frac{2^{2l}}{2^l}\right) = O(2^l) .$$

Now we can combine the multiscale block method and the overcomplete lifting scheme described in the previous section to achieve better performance. Figure 5 demonstrates the overcomplete lifting scheme with block algorithm applied to a sine signal with discontinuities near the middle.



(a) Traditional spatial convolution



(b) Multiscale block convolution

Figure 4. Difference between a linear convolution with zero-paddings in the kernel (a) and the multiscale block algorithm with four sub-regions (b).

4. Computational Cost

The computation of the computational costs of overcomplete DWT algorithms is different from those of decimated ones. In the decimated DWT algorithms, filter lengths remain unchanged as the analyzing level j changes, while the input lengths are decimated by 2^{j-1} where $j = 1, 2, \ldots, J$. However in undecimated DWT's the input length remains unchanged, while the filter lengths are upsampled by 2^{j-1}. In this section, we compute total costs of the various overcomplete DWT algorithms by counting all input points at each level of analysis with different input lengths.

Rioul and Duhamel [5] computed costs of some well known decimated DWT algorithms, such as the direct filter bank algorithm, the FFT-based split radix algorithm and the fast running short-length FIR filtering algorithm, in terms of filter length. Daubechies and Sweldens [1] derived the computational cost of the general lifting scheme. The criterion they chose was the number of multiplications and additions involved in the filtering. Figure 6 tells us that the lifting scheme is the most efficient than any other DWT algorithms up to the filter length $L = 34$.

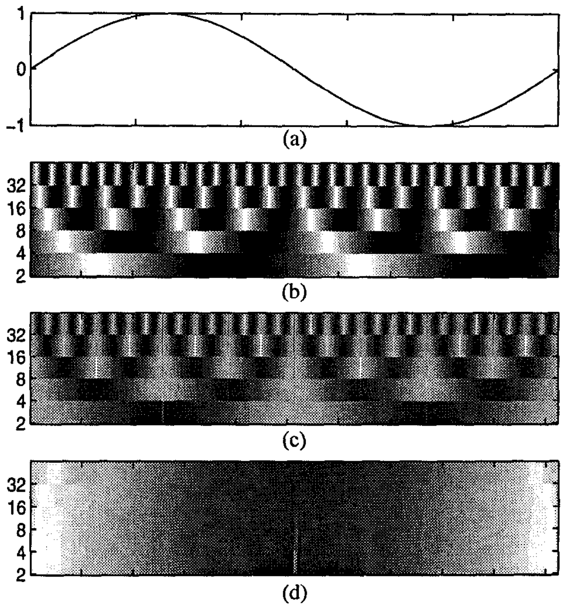


Figure 5. Implementation of the overcomplete lifting scheme with multiscale block algorithm (a) Original signal with discontinuities near the middle. (b) Blocked input signal. (c) Blocked wavelet coefficients after lifting. (d) Unblocked coefficients.

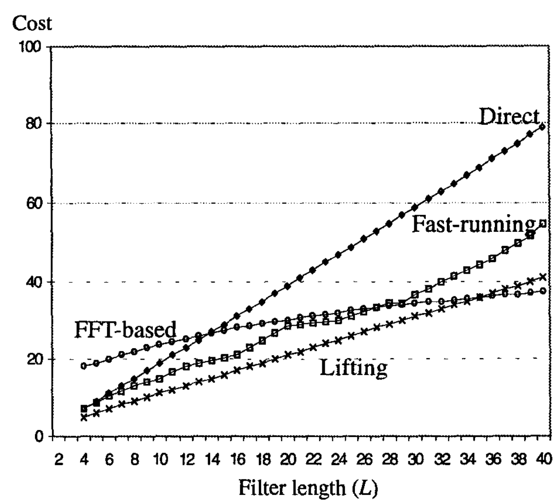


Figure 6. Comparison of the costs between various decimated DWT algorithms: direct filter bank algorithm, FFT-based split radix algorithm, fast running short-length filtering algorithm and lifting scheme.

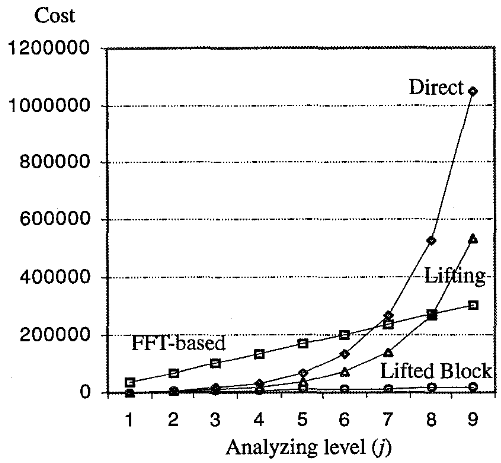


Figure 7. Costs of various overcomplete DWT algorithms with input size $N = 8$ and filter length $L = 3$.

However for some undecimated algorithms such as the direct filter bank algorithm and the lifting scheme, the filter length changes to $2^{j-1}(L-1)+1$ as the analyzing level j increases, while the FFT-based algorithms and the multiscale lifted block algorithm keep the filter length unchanged. If we denote by $C^j_{n,L}(X)$ the total cost of an overcomplete DWT algorithm X with 2^n input points at the j th octave with length-L filters, then

$$C^j_{n,L}(Direct) = 2^n(2^j(L-1)+1),$$

$$C^j_{n,L}(FFT\text{-}based) = 2^{n+1}(4n-3)+32,$$

$$C^j_{n,L}(Lifting) = 2^n(2^{j-1}(L-1)+2),$$

$$C^j_{n,L}(Lifting_{block}) = 2^n(L+1).$$

Figure 7 shows the computational costs of various overcomplete DWT algorithms, such as the direct filter bank algorithms, the FFT-based algorithms, the lifting scheme and the multiscale lifted block algorithm, with input size $n = 9$ and filter length $L = 3$ up to analyzing level $l = 9$.

5. Conclusion

In this paper we showed that the lifting scheme can be used for overcomplete multiscale wavelet representations with better performance without introducing boundary artifacts that exists in the traditional processing methods. Combined with the multiscale block algorithm, the new proposed lifting scheme helped overcomplete wavelet transforms to achieve an overall speedup of four over the traditional methods where an FFT is the basic implementation tool. We showed that the significantly improved performance of the proposed method makes undecimated wavelet transforms for feature analysis very advantageous in interactive processing paradigms such as a Web-based client-server model where the fast processing is highly desirable.

6. References

[1] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps", *Preprint*, Bell Laboratories, Lucent Technologies, 1996.

[2] G. Deslauriers, and S. Dubuc, "Symmetric iterative interpolating processes", *Constr. Approx.*, Vol 5, No. 1, pp. 49-68, 1989.

[3] D. Donoho, "Interpolating wavelet transforms", *Preprint*, Dept. Statistics, Stanford University, 1992.

[4] S. Mallat and S. Zhong, "Characterization of signals from multiscale edges", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, pp. 710-732, 1992.

[5] O. Rioul and P. Duhamel, "Fast Algorithms for Discrete and Continuous Wavelet Transforms", *IEEE Trans. on Inform. Theory*, Vol. 38, No. 2, pp. 569-586, 1992.

[6] M. Shim and A. Laine, "A fast algorithm to support interactive wavelet processing on a radiologist workstation," *Tech. Report* TR97-022, Dept. of Computer and Information Sciences and Engineering, University of Florida, 1997.

[7] W. Sweldens, "The Lifting Scheme: A New Philosophy in Biorthogonal Wavelet Constructions", *Wavelet Applications III*, A. Laine and M. Unser, Ed., Proc. SPIE, San Diego, CA, Vol 2569, pp. 68-79, 1995.

[8] W. Sweldens, "The Lifting Scheme: A custom-design construction of biorthogonal wavelets", *Appl. Comput. Harmon. Anal.*, Vol. 3, No. 2, pp. 186-200, 1996.