

# Shift-invariance in the Discrete Wavelet Transform

Andrew P. Bradley

Cooperative Research Centre for Sensor Signal and Information Processing,  
School of Information Technology and Electrical Engineering,  
The University of Queensland, St Lucia, QLD 4072, Australia  
{a.bradley@cssip.uq.edu.au}

**Abstract.** In this paper we review a number of approaches to reducing, or removing, the problem of shift variance in the discrete wavelet transform (DWT). We describe a generalization of the critically sampled DWT and the fully sampled *algorithme à trous* that provides approximate shift-invariance with an acceptable level of redundancy. The proposed over complete DWT (OCDWT) is critically sub-sampled to a given level of the decomposition, below which it is then fully sampled. The efficacy of the proposed algorithm is illustrated in an edge detection context and directly compared to a number of other shift-invariant transforms in terms of complexity and redundancy.

## 1 Introduction

The wavelet transform (WT) has been shown to be an invaluable tool in signal processing applications such as data compression and fast computations [8]. However, the most commonly used implementation of the WT: the critically sampled DWT [5], is shift variant and so is unsuitable for many signal analysis applications [6]. Therefore, to enable applications such as edge detection and texture analysis a number of redundant transforms have been proposed that are, at least approximately, shift-invariant [4,6,11,13]. In this paper we shall present an overview of some of these shift-invariant transforms and introduce a previously overlooked generalisation of the Mallat [5] and *à trous* [8] algorithms that provides a trade-off between the sparsity of the representation and shift-invariance. The proposed wavelet transform, which we shall refer to as the over complete discrete wavelet transform (OCDWT), is critically sub-sampled to a given level of the decomposition, below which it is then fully sampled. We shall demonstrate the efficacy of the OCDWT in an edge detection context and show that a sub-band can be made approximately shift-invariant provided that the next highest level of the decomposition was fully sampled.

In this paper, we shall refer to the WT in its broadest sense, that is, as a mathematical tool for multi-resolution analysis. However, to delineate the available implementations of the WT we shall distinguish [3]: The continuous wavelet transform (CWT) as a wavelet transform with a (suitably sampled) continuous-time mother wavelet, a continuous dilation (scale) parameter, and a discrete translation parameter; The discrete wavelet transform (DWT) as a wavelet transform with a discrete-time mother wavelet, (non-zero) integer dilation parameter, and a discrete translation parameter. To further distinguish different implementations of the DWT, we shall

reserve the use of the acronym DWT to refer to an (bi-) orthogonal WT. That is, a critically sampled DWT with a discrete-time mother wavelet applied at dyadic scales. We shall then refer to redundant (non-orthogonal) varieties of the DWT by pre-pending a suffix that highlights the type of redundancy introduced.

## 1.1 Shift-invariance and the DWT

The DWT is attractive primarily because the Mallat algorithm is a computationally efficient implementation of the WT and, depending on the mother wavelets used it is an (bi-) orthogonal transform. In particular, the DWT provides a sparse time-frequency representation of the original signal (the wavelet coefficients) that has the same number of samples as the original signal. Whilst an (bi-) orthogonal transform has been shown to be important for applications such as data compression and fast calculations, such importance has not been shown for multi-resolution analysis [7].

The most significant potential problem with the DWT is that it is a shift variant (i.e., not shift-invariant) transform [4,6,11,13,14]. Shift variance results from the use of critical sub-sampling (down-sampling) in the DWT. In this way, every second wavelet coefficient at each decomposition level is discarded. This is done both to reduce the amount of data that has to be analysed and to enforce the implicit time-frequency uncertainty of the analysis (as the analysis becomes more certain about the frequency components of the signal it becomes less certain about where they occur in time). This critical sub-sampling however, results in wavelet coefficients that are highly dependent on their location in the sub-sampling lattice. This can lead to small shifts in the input waveform causing large changes in the wavelet coefficients, large variations in the distribution of energy at different scales, and possibly large changes in reconstructed waveforms.

Another way of describing this phenomenon is to consider the frequency response of the mother wavelets. As no realisable wavelet filter (i.e., with compact support) can have an ideal “brick-wall” frequency response, i.e., the attenuation in the stop band will always be finite, aliasing will be introduced. That is, when the WT sub-bands, which nominally have half the bandwidth of the original signal, are sub-sampled by a factor of two, the Nyquist criteria is strictly violated and frequency components above (or below) the cut-off frequency of the filter will be aliased into the wrong sub-band (see [10], Chapter 4).

It should be noted that the aliasing introduced by the DWT cancels out when the inverse DWT (IDWT) is performed using *all* of the wavelet coefficients, that is, when the original signal is reconstructed. This makes a DWT followed by an IDWT shift invariant only when all of the wavelet coefficients are used to perform the IDWT, as soon as coefficients are not included in the IDWT, or they are quantised, the aliasing no longer cancels out and the output is no longer shift-invariant.

## 1.2 Shift-invariant Wavelet Transforms

There are a number of possible solutions to the shift variance problem, which we describe in more detail below. All of the techniques attempt to eliminate or minimise

the amount of aliasing that occurs by a combination of relaxing the critical sub-sampling criteria and/or by reducing the transition bandwidth of the mother wavelets.

The simplest way of making the DWT shift-invariant is not to perform any sub-sampling at all. This is most commonly referred to as the *algorithme à trous* [8]. Because there is no sub-sampling of data the mother wavelet has to be dilated (by inserting zeros) at each level of the transform. Obviously, the *à trous* algorithm is shift-invariant and it can be used with any of the mother wavelets conventionally used with the DWT. However, the *à trous* algorithm requires additional computation and memory and it is only strictly shift-invariant under circular convolution (periodic boundary extension).

Another WT, that performs no sub-sampling, is the continuous wavelet transform (CWT) [3,8]. In the CWT the mother wavelet is a continuous function, such as a *Mexican hat* wavelet (second derivative of a Gaussian). An advantage of the CWT is that it can be applied at any scale directly, without the iterations required by the DWT. In addition, there is no need (and in fact it often does not exist) to perform the inverse transform, i.e., the multi-resolution analysis is performed on the wavelet coefficients directly. This means that for a small number of analysis scales the CWT *may* be more computationally efficient than the DWT.

A WT that achieves approximate shift-invariance by limiting the sub-band sub-sampling is the power shiftable discrete wavelet transform (PSDWT) [13]. By designing the mother wavelet in the frequency domain to minimise sub-band aliasing and only sub-sampling at the second level of the transform and below the PSDWT meets a specified criteria of *power shiftable*. Power shiftable is defined so that although the wavelet coefficients in each sub-band may vary as the input signal shifts, the power in each sub-band remains constant. This definition of shift-invariant is suitable for applications such as texture analysis, but may not be strict enough for applications such as edge detection.

A more complex way of minimising shift variance is to build two wavelet decomposition trees (with alternate phase sub-sampling), one for a mother wavelet with even symmetry and the other for the same mother wavelet, but with odd symmetry. In this way, the dual tree complex wavelet transform (DTCWT) [4] measures both the real (even) and the imaginary (odd) components of the input signal (hence the name complex wavelet transform). The DTCWT is again approximately shift-invariant, and offers both magnitude and phase information. However, as two decompositions have to be performed computation and memory requirements are twice that of the Mallat DWT.

An alternative decomposition methodology which is not, strictly speaking, a WT (though it is related, see [3], Section 8.3.5) is the Laplacian pyramid (LP) [2]. The input signal is initially smoothed with a Gaussian filter and then down-sampled by a factor of two. This approximation signal is then up-sampled using nearest neighbour interpolation and then subtracted from the original signal. This difference signal (which is at the same resolution as the original signal) then defines the detail information lost during the smoothing, down-sampling, and up-sampling process. This process can be iterated a number of times to produce a sub-sampled low pass signal and a number of error signals equal to the number of levels of iteration. The original signal can be reconstructed by iteratively interpolating the low-pass signal and then adding the difference (detail) signal. The reason this transform is called the Laplacian pyramid is that the error signals, which are effectively a difference of Gaussian signals

(with the second Gaussian having half the bandwidth of the first) are indistinguishable from the second derivative, or Laplacian, of a Gaussian [9].

An additional method for obtaining a sub-sampled shift-invariant transform is via wavelet transform modulus maxima [8, Section 6.2, and 6]. A fully sampled dyadic WT, utilising a mother wavelet that estimates the first or second derivative of the signal (i.e., with a mother wavelet with one or two vanishing moments) is applied to estimate the multi-resolution gradient of the signal. This dyadic WT has the same properties as the CWT and so is shift-invariant. In addition, if the WT coefficients can be adaptively sub-sampled to keep only the coefficients that are locally maximum or locally minimum (the modulus maxima) at each scale, this sub-sampled representation is also shift-invariant. However, exact reconstruction from the wavelet modulus maxima, using a pseudo inverse transform, is not possible (the signal can only be recovered with a mean squared error of around  $10^{-2}$  [8]) and so it may not be suitable for applications that require perfect reconstruction.

## 2 The Over-complete DWT

It has been shown that the *à trous* and Mallat algorithms are both special cases of the same filter bank structure [12]. Therefore, in principle it is possible to combine both algorithms in the same decomposition and gain the benefits of both approaches. That is, the computational efficiency and sparse representation inherent in the critically sampled Mallat algorithm and the shift-invariance inherent in the fully sampled *à trous* algorithm. The approach presented in this paper is to apply the Mallat algorithm to the first  $M$  levels of an  $L$ -level decomposition and then apply the *à trous* algorithm to the remaining  $(L - M)$  levels. The proposed OCDWT algorithm can be viewed as an initial down-sampling of the signal prior to a fully sampled *à trous* decomposition. Alternatively, it be seen as a generalisation of the DWT, that produces the conventional DWT when  $M = L$  and produces the fully sampled *à trous* algorithm when  $M = 0$ .

Suppose that the scaling functions and wavelets  $\phi, \varphi, \hat{\phi},$  and  $\hat{\varphi}$  are designed with the filters  $h, g, \hat{h},$  and  $\hat{g}$  respectively. Now let  $a_0[n] = \int_{-\infty}^{\infty} f(t)\phi(t-n)dt$  be the (length  $N$ ) discrete samples formed from a local averaging of the continuous signal  $f(t)$  in the neighbourhood of  $t = n$ . For a filter  $h[n]$  we denote  $h_j[n]$  as the filter obtained by inserting  $2^j - 1$  zeros between each sample (tap) of  $h[n]$ , where  $j$  is the scale parameter, ( $j \in \mathbb{Z}$ ). This process of up-sampling dilates the mother wavelet, the Fourier transform of  $h_j[n]$  being  $H(2^j\omega)$  [8,10]. We also denote time reversal as,  $\bar{x}_j[n] = x_j[-n]$  and zero insertion as,

$$\tilde{x}[n] = \begin{cases} x[p] & \text{if } n = 2p \\ 0 & \text{if } n = 2p + 1 \end{cases} \quad (1)$$

The following convolution formulas are iterated to compute the dyadic OCDWT. For any  $j \geq 0, M \geq 0$ ,

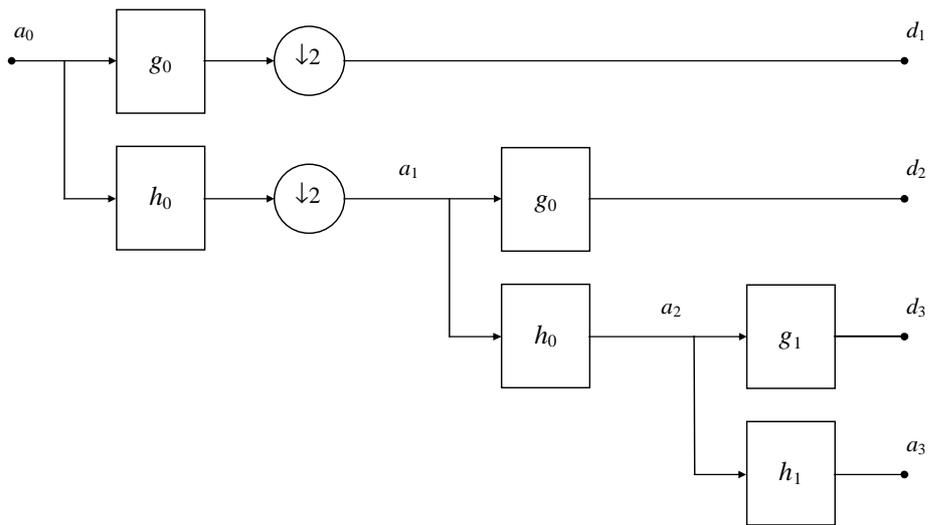
$$a_{j+1}[n] = \begin{cases} a_j * \bar{h}[2n] & j < M \\ a_j * \bar{h}_{j-M}[n] & \text{otherwise.} \end{cases} \quad \text{and} \quad (2)$$

$$d_{j+1}[n] = \begin{cases} a_j * \bar{g}[2n] & j < M \\ a_j * \bar{g}_{j-M}[n] & \text{otherwise.} \end{cases} \quad (3)$$

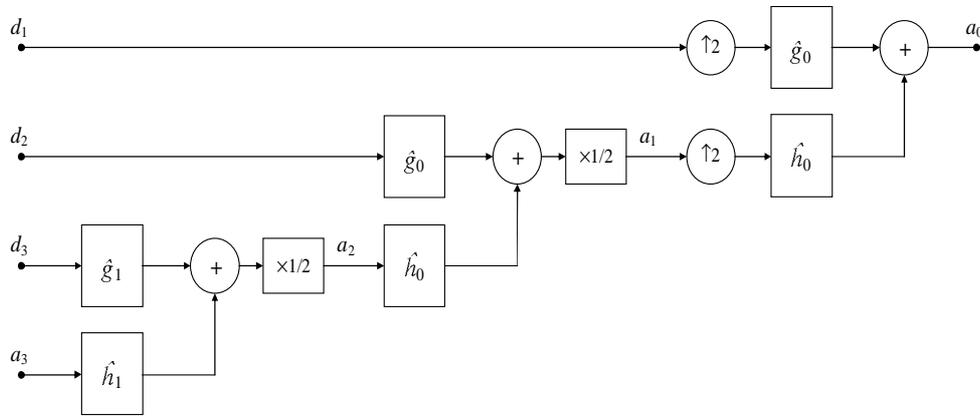
The inverse OCDWT is computed by iterating,

$$a_j[n] = \begin{cases} \check{a}_{j+1} * \hat{h}[n] + \check{d}_{j+1} * \hat{g}[2n] & j < M \\ \frac{1}{2}(a_{j+1} * \hat{h}_{j-M}[n] + d_{j+1} * \hat{g}_{j-M}[n]) & \text{otherwise.} \end{cases} \quad (4)$$

As an example, Fig. 1 illustrates an OCDWT analysis filter bank consisting of a single level of the Mallat algorithm and two levels of the *à trous* algorithm ( $L = 3, M = 1$ ), while Fig. 2 shows the (inverse) OCDWT synthesis filter bank used to reconstruct the original signal in this case.



**Fig. 1.** Forward three level OCDWT, one level critically sampled ( $L = 3, M = 1$ ).



**Fig. 2.** Inverse three level OCDWT, one level critically sampled ( $L = 3, M = 1$ ).

### 2.1 Computational Complexity

The computational complexity of an implementation of the WT is of particular interest in computer vision as the amount of data is large (say > 1 Mbytes) and the WT has to be applied both along the rows and down the columns of the image. Table 1 illustrates the memory requirements (in bytes, assuming 8 bit data) and computational complexity (as number of multiplications required) for a series of WT algorithms (forward transform only). In Table 1,  $N$  is number of samples in the one dimensional input signal;  $K_g$  and  $K_h$  are the number of non-zero coefficients in the high pass and low pass analysis filters respectively;  $L$  is the number of levels (scales) in the wavelet analysis; and  $M$  is the number of levels of the OCDWT that are critically sampled. It should be noted that the complexity will be doubled if we either want to perform the inverse transform to reconstruct a signal or perform a separable decomposition on two dimensional image data.

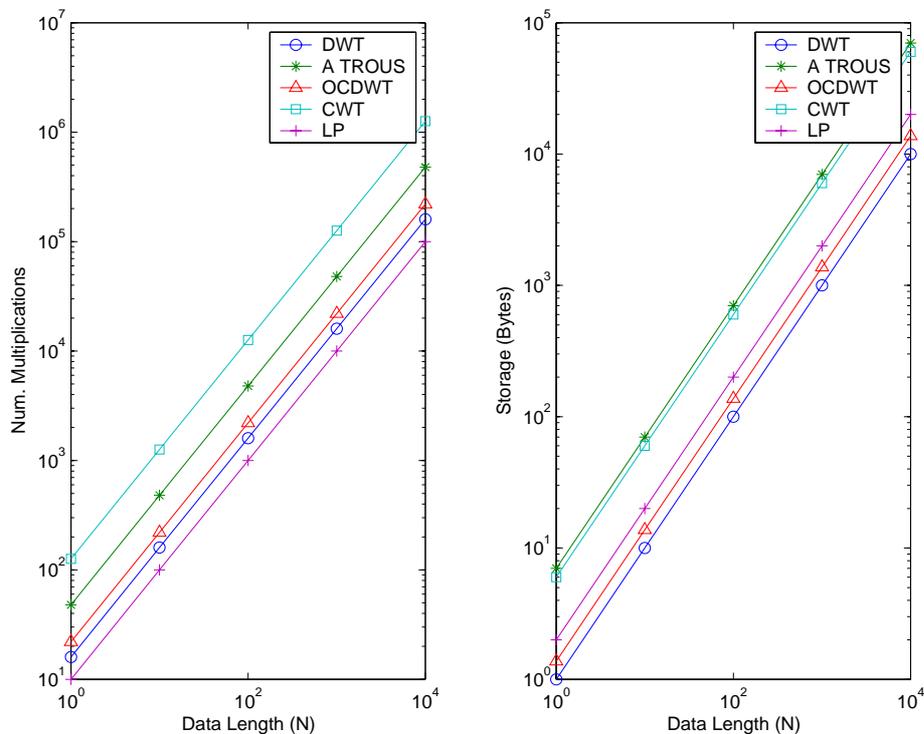
**Table 1:** Memory and complexity (number of multiplications) of various wavelet transform algorithms.

Decomposition	Memory	Complexity
DWT	$N$	$2(K_g + K_h)N$
A TROUS	$(L+1)N$	$L(K_g + K_h)N$
OCDWT	$N + (L - M)N/2^M$	$2(K_g + K_h)N + (L - M)(K_g + K_h)N/2^{(M-1)}$
CWT	$LN$	$L K_h N$
DTCWT	$2N$	$4(K_g + K_h)N$
LP	$2N$	$2K_h N$

As already mentioned in Section 1.2, one way to reduce the shift variance properties of a sub-sampled WT is to reducing the amount of sub-band aliasing. This is most simply done by increasing the number of vanishing moments of the mother wavelet used. However, this increases the support length of the wavelet filters, which

has a direct effect on computational complexity. Therefore, a combination of increasing filter length and reduced sub-sampling has to be used. It should also be noted that the computations quoted in Table 1 could be further reduced if advantage is taken of any symmetry properties of the wavelets or if any of the multiplications can be replaced with bit shifts.

Fig. 3 shows the computational complexity and storage requirements for a number of wavelet transform algorithms. These results are all based on a six level decomposition, three levels being critically sampled for the OCDWT ( $M = 3$ ), and the use of the 2<sup>nd</sup> order bi-orthogonal splines [3,8] for the DWT, *à trous*, and the OCDWT (length 3 and 5), second derivative of Gaussian filter (length 21) for the CWT [3], and a binomial filter (length 5) for the LP [2]. The DTCWT is not shown in Fig. 3,



**Fig. 3.** Computation and storage requirements for a number of WT algorithms.

Fig. 3 illustrates the trade-offs that are possible between computation and storage requirements between the various WT algorithms. For example, the most computationally efficient algorithm is not the most memory efficient. It also shows that, because of the long filter required, the CWT is the most computationally complex algorithm and requires approximately the same amount of storage as the fully sampled *à trous* algorithm. However, the *à trous* algorithm is less computationally demanding than the CWT and is therefore often preferred [12]. The LP requires the least number of multiplications as there is only one (low-pass) filter applied, the (high-pass) difference signal being generated using subtraction only. It is clear that both in terms of computation and storage the OCDWT will always fall somewhere between the *à*

*trous* and Mallat DWT algorithms. However, dependent upon the application, and input signal, the level of sub-sampling can be adjusted and traded-off against the required level of shift-invariance.

## 2.2 Application to Edge Detection

Points of sharp variations in intensity (edges) are often the most visually important features in an image [9]. The WT is a suitable tool for multiscale edge detection as it estimates the multiscale  $n^{\text{th}}$  order derivatives of a signal using mother wavelets with  $n$  vanishing moments [6]. In this section we briefly investigate the application of the OCDWT to multiscale edge detection. For illustration purposes we choose to detect edges as the zero crossings in the second derivative of the signal [9] rather than, the often preferred, maxima of the first derivative [6].

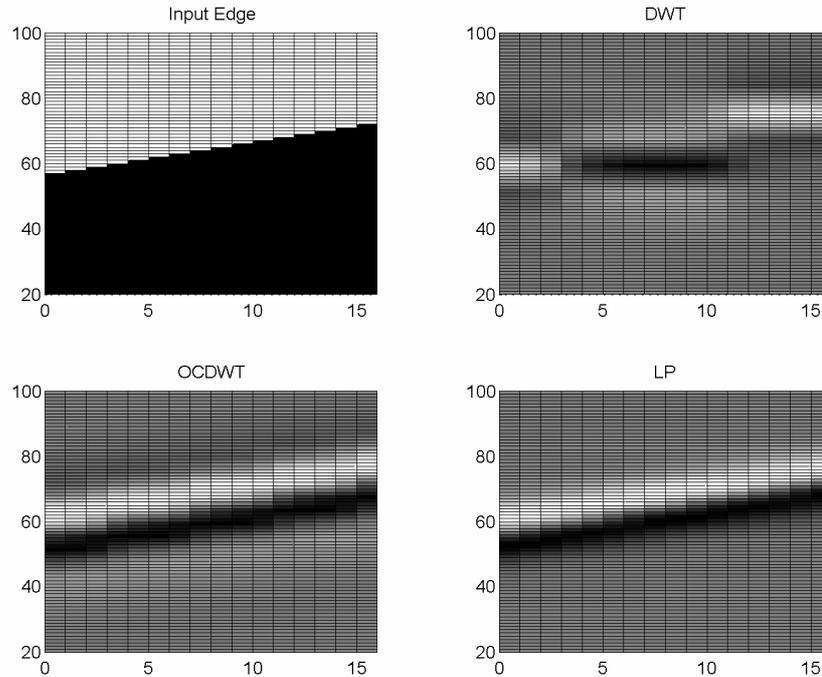
In order to evaluate the shift-invariance properties of a number of WT algorithms a one-dimensional input signal with a single discontinuity (step edge) was generated. This signal then became the columns of an *image*, with the location of the discontinuity being shifted by one sample along subsequent columns of the image. In this way a 16 pixel wide image, with a single linear edge was produced, as shown in Fig. 4. Various, 4-level, wavelet decompositions were then performed on the columns of the test image and reconstructions were generated using only coefficients from the level 4 low-pass (approximation, A4) and level 4 high-pass (detail, D4) sub-bands. Note, a 4-level dyadic transform was chosen as the sub-sampling lattice repeats every 16 samples. All results were generated using the same filters as used in Section 2.1, with the exception of the DTCWT which used length 13 and 19, near symmetrical filters. Finally the mean absolute error (MAE) was measured between the reconstructed signals at shift zero (column 1) and shifts 1 to 15 (columns 2 to 16). MAE was measured on the reconstructed signal, rather than the actual wavelet coefficients so as to normalise for different filter gains and sampling resolutions of the various WT algorithms.

**Table 2.** Shift-invariance properties of various WT algorithms.

WT Algorithm	Shift-invariance properties	MAE A4	MAE D4
CWT	Yes (circular convolution)	0.00	0.00
A TROUS	Yes (circular convolution)	0.00	0.00
OCDWT ( $M=1$ )	Approximate	0.06	0.14
LP	Approximate	0.20	0.24
OCDWT ( $M=2$ )	Approximate	0.24	0.64
DTCWT	Approximate	1.03	0.95
OCDWT ( $M=3$ )	Approximate	1.28	3.62
DWT	No (on sub-sampling lattice only)	6.81	8.06

Table 2 illustrates the shift-invariance properties of various WT algorithms. Results are presented in decreasing order of the degree of shift-invariance and they clearly show that the DWT is extremely *shift variant*. This result is also clearly illustrated in Fig. 4 as the location of the edge (zero crossing in the D4 reconstruction) varies unpredictably as the location of the discontinuity (edge) shifts. The OCDWT edge reconstruction was generated with two levels of critical sampling ( $M = 2$ ) and is

comparable to that of the LP for similar complexity and storage. Table 2 also shows that the MAE observed in the detail sub-bands are, in general, greater than that in the approximation sub-bands. This is due to the fact that the detail sub-bands have aliasing present from both from the lower (A4) and higher frequency sub-bands (D3).



**Fig. 4.** Level 4 detail reconstructions of a number of WT algorithms.

An additional advantage of the OCDWT is that it calculates a super-set of coefficients calculated by the DWT. Therefore, it is directly applicable where both the DWT and shift-invariance are required, e.g., as a visual model for guiding the quantisation of DWT coefficients in image compression. The shift-invariance provided by the OCDWT increases the reliability of detection of the smooth, edge, and textured regions so that spatially varying quantisation can be applied directly in the wavelet domain [1]. It may also be suitable for extracting the *primal sketch* of an image [9], which is commonly only meaningfully extracted from the larger scale edges, i.e., higher levels of WT, and so the lower levels can be critically sampled as it is not required that they are shift-invariant.

### 3 Conclusions

In this paper we have presented an overview of a generalisation of the Mallat and *à trous* discrete wavelet transform (DWT) algorithms, which we have termed the over complete discrete wavelet transform (OCDWT). The OCDWT achieves various levels of shift-invariance by controlling the amount of sub-sampling that is applied at each level of the transform. In this way, specific levels of the wavelet decomposition can be made approximately shift-invariant by ensuring that the level below is fully sampled (all lower sub-bands can be critically sub-sampled). The efficacy of the OCDWT has been demonstrated in an edge detection context and has been shown to be at least as shift-invariant as other WT algorithms with comparable computational complexity and redundancy.

### References

1. Bradley, A.P.: A Wavelet visible difference predictor, *IEEE Transactions on Image Processing*, Vol. 8, No. 5 (1999) 717-730.
2. Burt, P.J., and Adelson, E.H.: The Laplacian pyramid as a compact image code, *IEEE Transactions on Communications*, Vol. 31, No. 4 (1983) 532-540.
3. Daubechies, I.: Ten lectures on wavelets, Society for Industrial and Applied Mathematics. Philadelphia, Pennsylvania (1992).
4. Kingsbury, N.G.: Complex Wavelets and shift invariance, *Proceedings IEE Colloquium on Time-Scale and Time-Frequency Analysis and Applications*, London (2000).
5. Mallat, S.G.: A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 2 (1989) 674-693.
6. Mallat, S.G., and Zhong, S.: Characterisation of signals from multiscale edges, *IEEE Transactions of Pattern Analysis and Machine Intelligence*, Vol. 14, No. 7 (1992) 710-732.
7. Mallat, S.G.: Wavelets for a vision, *Proceedings of the IEEE*, Vol. 84, No. 4 (1996) 604-614.
8. Mallat, S.G.: *A Wavelet tour of signal processing*, 2<sup>nd</sup> Ed, Academic Press (1999).
9. Marr, D., and Hildreth, E.: Theory of edge detection. *Proceedings of the Royal Society London*, Vol 207 (1980) 187-217.
10. Oppenheim. A.V., Schafer, R.W., and Buck, J.R.: *Discrete-time signal processing*, 2<sup>nd</sup> Ed, Prentice Hall (1999).
11. Pesquet, J.C., Krim, H., and carfantan H.: Time-invariant orthonormal wavelet representations. *IEEE Transactions on Signal Processing*, Vol. 44, No. 8 (1996) 1964-1970.
12. Shensa, M.J.: The discrete wavelet transform: wedding the *à trous* and mallat algorithms. *IEEE Transactions on Signal Processing*, Vol. 40, No. 10 (1992) 2464-2482.
13. Simoncelli, E.P., Freeman, W.T., Adelson, E.H., and Heeger, D.J.: Shiftable multiscale transforms, *IEEE Transactions on Information Theory*, Vol. 38, No. 2 (1992) 587-607.
14. Unser M, and Aldroubi A.: A review of wavelets in biomedical applications. *Proceedings of the IEEE*, Vol. 84 (1996) 626-638.