Fast Image Operations in Wavelet Spaces *

Andrew Dorrell and David Lowe
School of Elec. Eng., University of Technology, Sydney
P.O. Box 123 Broadway 2007 Sydney Australia
email: {andrewd,dbl}@ee.uts.edu.au

Abstract

Wavelet transforms have been widely applied to image and video compression but they also have the ability to compress operators. A result is that there is great potential for producing fast algorithms for the processing of wavelet compressed image and video data. We examine the implementation of a set of fundamental image operations including pixel-wise arithmetic operators and arbitrary linear operators such as convolution.

1 Introduction

The computational expense associated with many image processing operations represents a significant obstacle to the integration of image and video information into many computer applications. This expense is a consequence of representing images as two dimensional arrays of intensity data—arrays which are large and dense (the majority of pixels are non constant). However image data is usually highly correlated. Appropriate linear transformation can remove much of this correlation; subsequent quantisation yielding a sparser representation; and these two steps form the basis of most image compression algorithms. By performing operations on this sparse representation, significant computational savings may be possible. This saving comes about through two means: firstly, the reduced volume of image (video) data can significantly reduce demand on systems resources [12]; secondly, the representation of certain operators in the transform space is also sparse—allowing the use of sparse matrix techniques [2, 1]. For image processing applications it is, additionally, possible that storage and processing of the data use the same (sparse) representation. This avoids computation of the transform which, ideally, need only be performed during display or acquisition (where it can be assigned to dedicated hardware). A first pass at this idea is given in [12, 11] based on a block Discrete Cosine Transform (DCT), allowing the use of standard (JPEG) compressed data.

There are a number of problems with using the block DCT representation for image processing however. Of particular concern to us is the poor representation of scale information. Scale is a fundamental property of image data important to a wide range of image processing tasks (eg. edge detection). As a result, mapping of these tasks to the JPEG domain may be be complex—reducing the advantage of sparse representation. Algorithms with real-time constrains (such as the processing and display of video) can also be disadvantaged by the lack of scalability (in the sense of [3]) of the block DCT representation. The problem here is that reconstruction at reduced resolution may not provide a corresponding saving in computation or bandwidth. For these reasons, the use of a wavelet based representation is more desirable.

*Presented at the biannual conference of the Australian Pattern Recognition Society (APRS)—DICTA95 (Digital Image Computing Techniques and Applications), Brisbane Australia.
Wavelet based image and video representations explicitly separate information at different scales and are closely related to the pyramidal structures found in image processing literature [7]. Their performance in compression, progressive coding, and processing applications make them ideally suited for use by computer applications. However, while wavelet space image analysis has been widely reported (motion estimation [4], texture analysis [5], contrast enhancement [9]), the mapping of image space operators to wavelet spaces has only received significant attention in the mathematical (numerical analysis) literature (e.g. [2, 1]). As a result, little image processing work makes explicit use of these techniques. On the other hand, the numerical literature does not fully address a number of image specific considerations.

In this paper, we aim to present to the image processing community, a number of algorithms for the processing of data represented in a wavelet space. In section 2.1, we briefly introduce the wavelet transform before extending the work of [12, 11] to the wavelet domain in section 2.2. The convolution operator is examined in more detail in section 3 and a new algorithm for its fast implementation is suggested. In section 4 orthogonal, biorthogonal and over-complete wavelet transforms are discussed and compared for the purpose of implementing image processing tasks. Finally, section 5 presents some conclusions and describes the direction of ongoing research.

2 Theory

2.1 Wavelet Transforms

The wavelet transform [7, 6], as with many other common image transforms, may be viewed as a mapping of image data from Cartesian space to a function space. The basis functions of the wavelet transform are translations and dilations of a prototype function \( \psi \) called the mother wavelet (which may be associated with a bandpass filter). The one dimensional discrete wavelet transform of a function \( f(x) \) is defined

\[
W_{\psi}^{m,n} \{ f \} = |a_0|^{-m/2} \int f(t) \psi \left( a_0^{-m} t - n b_0 \right) dt
\]

(1)

where \( a \) and \( b \) define the discrete scale and translation step size and \( n \) and \( m \) are the discrete scale and translation variables.

The discrete wavelet transform is most commonly implemented as a cascaded filter bank (following the work of Mallat [10]) making it similar to subband coding—but with each subband having a bandwidth proportional to its centre frequency. In this work we will use the term subband to refer to the wavelet coefficients corresponding to a single filter output. In the two dimensional dyadic wavelet transform, the transform coefficients consist of a single low-pass (residual) subband and three directed (horizontal, vertical and diagonal) band-pass subbands for each level \( m \) of the transform. Where the wavelet basis is orthogonal or biorthogonal, the subbands may be critically down-sampled. In this case the number of output points is equal to the number of input points. Figure 1 shows an orthogonal, critically subsampled wavelet transform of an artificial test image to three levels along with the regions of the spatial frequency plane covered by each subband.

2.2 Mapping operators

We begin the consideration of mapping operators to the space defined by the wavelet transform by examining some of the inherent properties of the transform operator. The definition of the wavelet transform endows it with the following properties:

\[
W \{ f_1 + f_2 \} = W \{ f_1 \} + W \{ f_2 \}
\]

(2)

\[
\alpha f \Leftrightarrow \alpha W \{ f \}
\]

(3)

\[
O p \{ W X \} \Leftrightarrow (O p W) X,
\]

(4)
Figure 1: A two dimensional wavelet transform: (a) original image; (b) its wavelet transform performed to three levels using Daubechies \( D_4 \) wavelet (subbands are independently normalised to enhance contrast). The subbands cover the regions of the spatial frequency plane indicated by the corresponding (square) region in (c).

where \( f_1 \) and \( f_2 \) are functions, \( W \) denotes the wavelet transform, \( \alpha \) is a scalar and \( Op \) is some arbitrary linear operator. That is, the wavelet transform is distributive over addition (2); a scaling of a function produces a corresponding scaling of its wavelet coefficients (3); and the transform operator is associative with other linear operators. The wavelet transform shares these properties with the Fourier transform and other related (linear) transforms such as the DCT. These three basic properties provide the basis for much of the work reported in [12] which we will now extend to the wavelet domain.

### 2.2.1 Some simple operations

Equations (2–4) result in trivial definitions for a number of simple arithmetic operations on images. In particular, (3) says we may scale the pixels in an image (by some scalar \( \alpha \) by scaling the wavelet coefficients. Similarly, by (2), two images may be added (pixel-wise) by adding their wavelet coefficients. As with image space addition, this is dependent on the operand images being the same size. In addition they must be decomposed in the same wavelet basis and to the same number of levels.

A definition for scalar addition follows in a relatively straightforward way by observing that scalar addition is equivalent to pixel-wise addition where the second operand is an image of constant value. The wavelet transform of such an image has constant valued subbands with vanishingly small values in all except the low-pass subband. Significant computational saving can therefore be made by scaling only the low-pass subband. For the infamous \( 512 \times 512 \) Lenna image, normalised to 256 grey levels then wavelet decomposed to four levels using a Daubechies \( D_4 \) wavelet, the RMS error introduced by this technique was in the order of \( 10^{-5} \) (the same order as the numerical error introduced during forward and inverse transforms)—for a 64-fold reduction in computation.

It should be noted that in many implementations of the wavelet transform, the subbands are not scaled after filtering. In this case the scalar operand must be scaled by a factor of \( |\alpha|^{-1/2} \) prior to being added to the low-pass subband.

### 2.2.2 Arbitrary linear operators

To determine a mapping of a linear operator to the wavelet space we need to use the associativity property of the (wavelet) transform (4). First, let \( \mathcal{W}_\psi \) be the wavelet transform
operator with inverse $W_{\psi}^{-1}$; $H$ is a linear operator; and $x$ and $y$ are images, with wavelet coefficients $\hat{x}$ and $\hat{y}$, such that $y = Hx$. The sequence of operations required to apply the operator $H$ to $\hat{x}$, producing $\hat{y}$ is

$$\hat{y} = W_{\psi}H W_{\psi}^{-1} \hat{x}.$$  

That is, apply the inverse transform to $\hat{x}$, apply $H$ the the result, then apply the forward transform to this result. By the associativity property (4), we may regroup these operations as

$$\hat{y} = \left(W_{\psi}H W_{\psi}^{-1}\right) \hat{x},$$

$$\hat{y} = \hat{H} \hat{x}.$$  

Then the new operator $\hat{H} = \left(W_{\psi}H W_{\psi}^{-1}\right)$ is a representation of the operator $H$ in the wavelet space.

The method for transforming the operator $H$ is best described for the one dimensional case. For the case where $x$ and $y$ are 1D signals, the operator $H$ may be represented as a matrix. This matrix is applied to $x$ by multiplication. The standard method for calculating $\hat{H} = \left(W_{\psi}H W_{\psi}^{-1}\right)$ is to apply 1D transformation to each row of $H$ and then to each column of the result. The matrix which results is referred to as the “standard form” of the operator [2]. Due to the well defined structure of this matrix it is possible to apply it in $O(N)$ operations (where $N$ is the length of the signal to which it is applied). The mapping of the operator to the wavelet space is, in general an $O(N^2)$ problem, though, if $H$ is well structured, it may be performed in $O(N)$ [2]. This expense becomes insignificant if the operator is to be applied repeatedly as would be the case when processing a video stream.

The 1D formulation extends in a straightforward way to image (and video) data where the operator is separable. The resulting complexity is $O(N^D)$ where $D$ is the dimension of the data. Additionally, it is possible to perform some truncation of the operator in order to reduce the computation further. In this case, similar quantisation techniques to those used in image compression may be applied to the transformed operator.

3 On Convolution

Convolution represents a special case of a linear operator with particular importance to image processing. An example of a convolution operator and its wavelet space representation is given in figure 3. The most straightforward method for applying this operator to our data is using matrix multiplication (as depicted in figure 2). Although this can be performed efficiently using sparse matrix techniques, the method exploits little of the well defined structure.

![Figure 2: Structure and application of a linear operator in the wavelet space. $s_m$ is the low-pass subband, $d_m$ the band-pass subbands. The sub-matrices $A_{ij}, B_{ij}$ and $C_{ij}$ describe the interrelations between input subband $i$ and output subband $j$—i.e. the shifting of information between subbands as a result of the operation.](image)
of the matrix. In this next section we indicate briefly how this structure may be exploited to yield an algorithm for the application of the operator by subband convolution. The algorithm will be described more fully in [8].

The transformed convolution operator of figure 3b has a fingered structure which results from the diagonal dominance of each of the sub-matrices $A_i$, $B_{i,j}$ and $C_{i,j}$. As each of these sub-matrices acts on a single subband of the input data, we may postulate that it could be implemented using a convolution. For this to be the case, each row would need to be a shifted version of the other rows of the matrix. We find this to be the case for the sub-matrices $A_i$ and $T_i$. It turns out that this is also the case for the sub-matrices $C_{i,j}$ but in this case, each row is shifted by $\text{width}/\text{height}$ of the sub-matrix. That is, the inter-row shift in $C_{i,2}$ is 2; the inter-row shift in $C_{4,1}$ is 4; and so on. Such a matrix may be applied efficiently as a decimated convolution.

The more complex case is that of the sub-matrices $B_{i,j}$. It can be shown that the rows of each sub-matrix may be composed from shifted versions of $\text{height}/\text{width}$ different functions. These are interlaced so that a group of functions appear as a block which recur along the main diagonal. The bottom graph of figure 3c depicts the case where each block is composed of two different functions. Each of these functions may therefore be applied as a separate convolution with the result being appropriately up-sampled and shifted.

By exploiting the above observations it is possible to minimise the storage required for the representation of the operator and provide a fast algorithm based entirely on convolution.

4 Discussion

The numerical algorithms work in [2, 1] makes use of (Daubechies) orthogonal compactly supported wavelets. In general however, a wide choice is available in the selection of mother wavelet. For image coding applications, where linear phase is important, it is most common to use biorthogonal wavelets as orthogonal wavelets cannot be symmetric. Symmetric wavelets, in turn, lend themselves to an alternative treatment of image boundaries by symmetric extension. This method reduces the discontinuity seen by the transform process compared to periodisation. The effect of symmetric wavelets and edge extension on the representation of wavelet space operators has yet to be addressed in the numerical literature.

The type of wavelet basis used may also effect the way certain operations can be implemented in the wavelet space. Consider for example the simple task of rotating an image by a
multiple of 90°. The algorithm which is the most intuitive to implement would simply rotate
the coded subbands prior to reconstruction. For this to work however, the basis must be
symmetric. If even length filters are used, delays must also be accounted for prior to recon-
struction. A similar algorithm for rotation by an arbitrary angle θ would require resampling
of the wavelet space. This, in general, can only be performed in the over-complete wavelet
space due to the shift variance of the maximally decimated coefficients.

5 Conclusions

Performing image operations in the wavelet space can provide significant computational
advantage. This is maximised where data is stored in a wavelet compressed format and
coding (decoding) is performed only for display (acquisition) purposes. In this paper we
have provided the implementation of a number of common image operations in the wavelet
space including simple pixelwise arithmetic and arbitrary linear operators. We also examined
the case of convolution in more detail, suggesting a fast algorithm for its implementation in
the wavelet space. The approach has greatest potential in the processing of video sequences
where an image size remains constant and, therefore, the wavelet space representation of
the operator need only be calculated once for application to many images (frames). In
future work we hope to make more explicit the effect of both symmetric edge handling and
redundant transform on the representation of operators in wavelet spaces. In addition, some
objective evaluation of the visual effects of operator compression is being undertaken.

References

ftp.colarado.edu/pub/wavelets/papers/ddx.ps.Z.
ftp://pascal.math.yale.edu/pub/wavelets/bcr.tex.
wavelet multiresolution analysis method for computing optical flow. Optical Engineer-
[10] S. Mallat. A theory for multiresolution signal decomposition: The wavelet represent-
tion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11(7):674–693,
July 1989.
sity.