

The Dual-Tree Complex Wavelet Transform – A *Coherent* Framework for Multiscale Signal and Image Processing

Ivan W. Selesnick

Electrical and Computer Engineering
Polytechnic University
6 Metrotech Center, Brooklyn, NY 11201, USA
Email: selesi@poly.edu, Web: taco.poly.edu/selesi/

Richard G. Baraniuk

Electrical and Computer Engineering
Rice University
MS-380, 6100 Main Street, Houston, TX 77005, USA
Email: richb@rice.edu, Web: dsp.rice.edu

Nick Kingsbury

Signal Processing Group, Department of Engineering
Cambridge University
Trumpington Street, Cambridge CB2 1PZ, UK
Email: ngk@eng.cam.ac.uk, Web: www-sigproc.eng.cam.ac.uk/~ngk/

Submitted to *IEEE Signal Processing Magazine*, July 2004

Revised

July 31, 2005

Abstract

The dual-tree complex wavelet transform (CWT) is a relatively recent enhancement of the discrete wavelet transform (DWT) with important additional properties: It is nearly shift-invariant and directionally selective in two and higher dimensions. It achieves this with a redundancy factor of only 2^d for d -dimensional signals, which is substantially lower than the undecimated DWT. The multi-dimensional dual-tree CWT is non-separable but is based on a computationally efficient, separable filter bank. This tutorial discusses the theory behind the dual-tree transform, shows how complex wavelets with good properties can be designed, and illustrates a range of applications in signal and image processing.

1 Introduction

1.1 The wavelet transform and multiscale analysis

Since its emergence twenty years ago, the *wavelet transform* has been exploited with great success across the gamut of signal processing applications, in the process often redefining the state-of-the-art of performance [102, 112]. In a nutshell, the discrete wavelet transform (DWT) replaces the infinitely oscillating sinusoidal basis functions of the Fourier transform with a set of *locally* oscillating basis functions, called *wavelets*. In the classical setting, the wavelets are stretched and shifted versions of a fundamental, real-valued bandpass wavelet $\psi(t)$. When carefully chosen and combined with shifts of a real-valued lowpass *scaling function* $\phi(t)$, they form

an orthonormal basis expansion for one-dimensional (1-D) real-valued continuous-time signals [27]. That is, any finite-energy analog signal $x(t)$ can be decomposed in terms of wavelets and scaling functions via

$$x(t) = \sum_{n=-\infty}^{\infty} c(n) \phi(t-n) + \sum_{j=0}^{\infty} \sum_{n=-\infty}^{\infty} d(j,n) 2^{-j/2} \psi(2^j t - n). \quad (1)$$

The scaling coefficients $c(n)$ and wavelet coefficients $d(j,n)$ are computed via the inner products

$$c(n) = \int_{-\infty}^{\infty} x(t) \phi(t-n) dt, \quad (2)$$

$$d(j,n) = 2^{-j/2} \int_{-\infty}^{\infty} x(t) \psi(2^j t - n) dt. \quad (3)$$

They provide a *time-frequency analysis* of the signal by measuring its frequency content (controlled by the scale factor j) at different times (controlled by the time shift n).

There exists a very efficient, $O(N)$ algorithm to compute the coefficients $c(n)$ and $d(j,n)$ from a fine-scale representation of the signal (often simply N samples) and vice versa based on two octave-band, discrete-time *filter banks* that recursively apply a discrete-time lowpass filter $h_0(n)$, a high-pass filter $h_1(n)$, and upsampling and downsampling operations (see Figure 24) [27, 69]. These filters provide a convenient parametrization for designing wavelets and scaling functions with desirable properties, such as compact time support and fast frequency decay (to ensure the analysis is as local as possible in time-frequency) and orthogonality to low-order polynomials (“vanishing moments”) [27]. See Sidebar A for more background on wavelets, filter banks, and their design.

Why have wavelets and multiscale analysis proved so useful in such a wide range of applications? The primary reason is because they provide an extremely efficient representation for many types of signals that appear often in practice but are not well matched by the Fourier basis, which is ideally meant for periodic signals. In particular, wavelets provide an optimal basis for signals containing *singularities* (jumps, spikes, and so forth), the archetypal example being a piecewise smooth function consisting of low-order polynomials separated by jump discontinuities. The wavelet representation is optimally *sparse* for such signals, requiring an order of magnitude fewer coefficients than the Fourier basis to approximate within the same error. The key to the sparsity is that since wavelets oscillate locally, only wavelets overlapping a singularity have large wavelet coefficients; all other coefficients are small.

The sparsity of the wavelet coefficients of many real-world signals enables near-optimal signal processing based on simple *thresholding* (“keep the large coefficients and kill the small ones”), the core of a host of powerful image compression (JPEG2000 [98]), denoising, approximation, and deterministic, and statistical signal and image algorithms.

1.2 Trouble in paradise: Four problems with real wavelets

But this is not the end of the story. In spite of its efficient computational algorithm and sparse representation, the wavelet transform suffers from four fundamental, intertwined shortcomings.

Problem 1 – Oscillations: Since wavelets are bandpass functions, the wavelet coefficients tend to oscillate positive and negative around singularities (see Figures 1 and 2, for example). This considerably complicates wavelet-based processing, making singularity extraction and signal modeling in particular very challenging [22]. Moreover, since an oscillating function passes often through zero, we see that the conventional wisdom that “singularities yield large wavelet coefficients” is overstated. Indeed, as we see in Figure 1 it is quite possible for a wavelet overlapping a singularity to have a small or even zero wavelet coefficient.

Problem 2 – Shift variance: A small shift of the signal greatly perturbs the wavelet coefficient oscillation pattern around singularities (see Figure 2). Shift variance also complicates wavelet-domain processing; algorithms must be made capable of coping with the wide range of possible wavelet coefficient patterns caused by shifted singularities [34, 55, 59, 80, 83].

To better understand wavelet coefficient oscillations and shift variance, consider a piecewise smooth signal $x(t - t_0)$ like the step function

$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t \geq 0 \end{cases}$$

analyzed by a wavelet basis having a sufficient number of vanishing moments. Its wavelet coefficients consist of samples of the *step response* of the wavelet.¹

$$d(j,n) \approx 2^{-3j/2} \Delta \int_{-\infty}^{2^j t_0 - n} \psi(t) dt,$$

where Δ is the height of the jump. Since $\psi(t)$ is a bandpass function that oscillates around zero, so does its step response $d(j,n)$ as a function of n (recall Figure 1).

¹This formula is exact for a piecewise constant signal [80, 83].

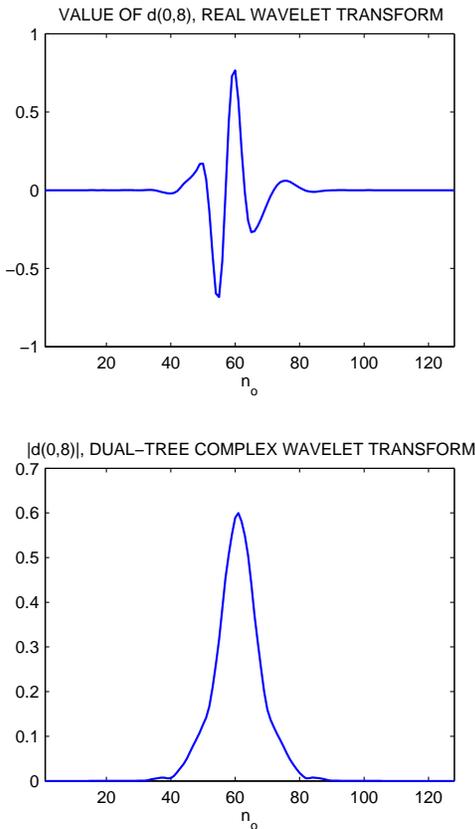


Figure 1: In the neighborhood of an edge, the real DWT produces both large and small wavelet coefficients. In contrast, the (approximately) analytic \mathbb{C} WT produces coefficients whose magnitudes are more directly related to their proximity to the edge. Here, the test signal is a step edge at $n = n_o$, $x(n) = u(n - n_o)$. The figure shows the value of the wavelet coefficient $d(0, 8)$ (the 8th coefficient at stage 3 in Figure 24) as a function of n_o . In the top panel, the real coefficient $d(0, 8)$ is computed using the conventional real DWT. In the lower panel, the complex coefficient $d(0, 8)$ is computed using the dual-tree \mathbb{C} WT. (The filters used here are the same as those in Figure 2).

Moreover, the factor 2^j in the upper limit ($j \geq 0$) amplifies the sensitivity of $d(j, n)$ to the time shift t_0 , leading to strong shift variance.

Problem 3 – Aliasing: The wide spacing of the wavelet coefficient samples, or equivalently the fact that the wavelet coefficients are computed via iterated discrete-time downsampling operations interspersed with non-ideal lowpass and highpass filters, results in substantial *aliasing*. The inverse DWT cancels this aliasing, of course, but only if the wavelet and scaling coefficients are not changed. Any wavelet coefficient processing (thresholding, filtering, quantization, and so on) upsets the delicate balance between the forward and inverse transforms, leading to artifacts in the reconstructed signal.

Problem 4 – Lack of directionality: Finally, while Fourier sinusoids in higher dimensions correspond to highly directional plane waves, the standard tensor product construction of multi-dimensional wavelets produces a “checkerboard” pattern that is simultaneously oriented along several directions. This lack of *directional selectivity* greatly complicates modeling and processing of *geometric* image features like ridges and edges. (More on this in Section 4 below.)

1.3 One solution: Complex wavelets

Fortunately, there is a simple solution to these four DWT shortcomings. The key is to note that the *Fourier transform* does not suffer from these problems. First, the magnitude of the Fourier transform does not oscillate positive and negative but rather provides a smooth positive envelope in the Fourier domain. Second, the magnitude of the Fourier transform is perfectly shift invariant, with a simple linear phase offset encoding the shift. Third, the Fourier coefficients are not aliased and do not rely on a complicated aliasing cancellation property to reconstruct the signal. And fourth, the sinusoids of the multi-dimensional Fourier basis are highly directional plane waves.

What is the difference? Unlike the DWT, which is based on *real-valued* oscillating wavelets, the Fourier transform is based on *complex-valued* oscillating sinusoids

$$e^{j\Omega t} = \cos(\Omega t) + j \sin(\Omega t) \quad (4)$$

with $j = \sqrt{-1}$. The oscillating cosine and sine components (the real and imaginary parts, respectively) form a *Hilbert transform pair*; that is, they are 90° out of phase with each other. Together they constitute an *analytic signal* $e^{j\Omega t}$ that is supported on only one-half of the frequency axis ($\Omega > 0$). See Sidebar B for more background on the Hilbert transform and analytic signals.

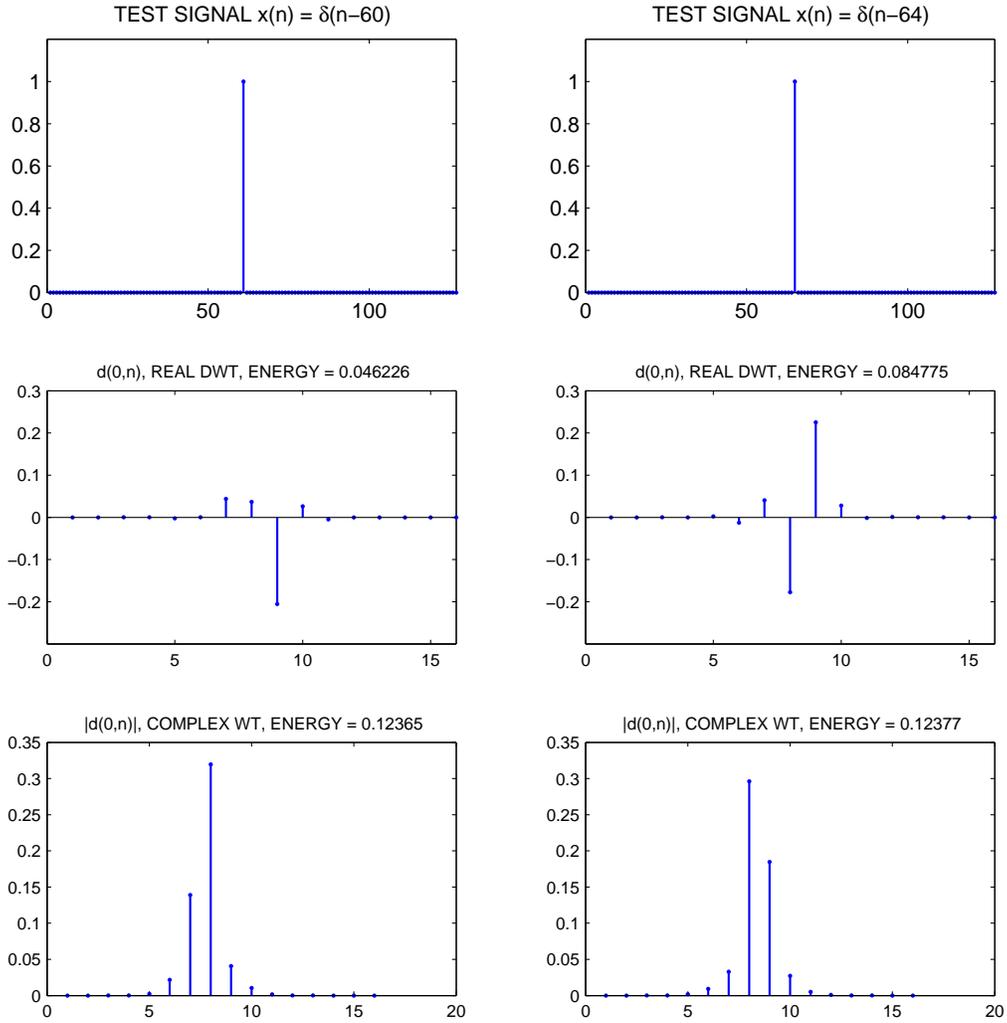


Figure 2: The wavelet coefficients of a signal $x(n)$ are very sensitive to translations of the signal. For two impulse signals $x(n) = \delta(n - 60)$ and $x(n) = \delta(n - 64)$ (top panel), we plot the wavelet coefficients $d(j, n)$ at a fixed scale j (middle and lower panels). The middle panel shows the real coefficients computed using the conventional real discrete wavelet transform (DWT, with Daubechies length-14 filters). The lower panel shows the magnitude of the complex coefficients computed using the dual-tree complex discrete wavelet transform (\mathbb{C} WT with length-14 filters from [58]). For the dual-tree \mathbb{C} WT the total energy at scale j is nearly constant, in contrast to the real DWT.

Inspired by the Fourier representation, imagine a *complex wavelet transform* (CWT)² as in (1)-(3) but with a complex-valued scaling function and complex-valued wavelet

$$\psi_c(t) = \psi_r(t) + j\psi_i(t).$$

Here, by analogy to (4), $\psi_r(t)$ is real and even and $j\psi_i(t)$ is imaginary and odd. Moreover, if $\psi_r(t)$ and $\psi_i(t)$ form a Hilbert transform pair (90° out of phase with each other), then $\psi_c(t)$ is an analytic signal and supported on only one-half of the frequency axis. The complex scaling function is defined similarly. See Figure 9 for an example of a complex wavelet pair that approximately satisfies these properties.

Projecting the signal onto $2^{-j/2}\psi_c(2^j t - n)$ as in (3), we obtain the *complex wavelet coefficient*

$$d_c(j, n) = d_r(j, n) + j d_i(j, n)$$

with magnitude

$$|d_c(j, n)| = \sqrt{[d_r(j, n)]^2 + [d_i(j, n)]^2}$$

and phase

$$\angle d_c(j, n) = \arctan\left(\frac{d_i(j, n)}{d_r(j, n)}\right)$$

when $|d_c(j, n)| > 0$. As with the Fourier transform, complex wavelets can be used to analyze and represent both real-valued signals (resulting in symmetries in the coefficients) and complex-valued signals. In either case, the CWT enables new *coherent multiscale signal processing algorithms* that exploit the complex magnitude and phase. In particular, as we will see, a large magnitude indicates the presence of a singularity while the phase indicates its position within the support of the wavelet [81, 83, 113, 117].

The theory and practice of discrete complex wavelets can be broadly classed into two schools. The first seeks a $\psi_c(t)$ that forms an orthonormal or biorthogonal basis [9, 11, 37, 64, 108, 114]. As we show below in Section 2.3, this strong constraint disables the resulting CWT from overcoming most of the four DWT shortcomings outlined above. The second school seeks a *redundant* representation, with both $\psi_r(t)$ and $\psi_i(t)$ individually forming orthonormal or biorthogonal bases. The resulting CWT is a $2\times$ redundant *tight frame* [26] in 1-D with the power to overcome the four shortcomings.

In this paper, we will focus on a particularly natural approach to the second, redundant type of CWT — the *dual-tree* approach — which is based on two filter bank (FB) trees and thus two bases [55, 57]. As we will see, any CWT based on wavelets of compact support cannot exactly possess the Hilbert transform / analytic signal properties, and this means that any such CWT will not perfectly overcome the four DWT shortcomings. The key challenge in dual-tree wavelet design is thus the joint design of its two FBs to yield a complex wavelet and scaling function that are as close as possible to analytic. From Figure 9, we see that we can reach quite close to the ideal even with quite short filters.

As a result, the dual-tree CWT comes very close to mirroring all the attractive properties of the Fourier representation, including a smooth, non-oscillating magnitude (see Figure 1); a nearly shift-invariant magnitude with a simple near-linear phase encoding of signal shifts; substantially reduced aliasing; and directional wavelets in higher dimensions. The only cost for all of this is a moderate redundancy: $2\times$ redundancy in 1-D (2^d for d -dimensional signals, in general). This is much less than the $\log_2 N \times$ redundancy of a perfectly shift-invariant DWT [23, 63], which moreover will not offer the desirable magnitude/phase interpretation of the CWT, nor the good directional properties in higher dimensions.

1.4 Paper organization

This paper aims to reach two different audiences. The first is the wavelet community, many members of which are unfamiliar with the utility, convenience, and unique properties of complex wavelets. The second is the broader class of signal processing folk who work with applications where the DWT has proved somewhat disappointing, such as those involving complex or modulated signals (radar, speech, and music, for example) or higher-dimensional, geometric data (geophysics and imaging, for example). In these problems, the CWT can potentially offer significant performance improvements over the DWT.

Section 2 of the paper describes the challenges in developing complex wavelet transforms. Section 3 introduces the dual-tree approach, overviews the design issues, and synthesizes three different solution approaches. Section 4 explains how to extend the dual-tree approach to construct real and complex directional wavelets for multi-dimensional geometric data. Section 5 deals with the use of complex wavelets through several real and stylized applications. While our aim is not to provide an exhaustive treatment of the myriad types of CWTs, we provide a brief overview of related techniques in Section 6. Section

²We use the complex number symbol \mathbb{C} in CWT to avoid confusion with the oft-used acronym CWT for the (different) continuous wavelet transform.

7 closes with conclusions. Finally, two sidebars on the DWT, the Hilbert transform and analytic signals provide background information for the development.

2 Complex Wavelet Complexities

The design of complex analytic wavelets raises several unique and nontrivial challenges that do not arise with the real DWT. In this section, we overview them and discuss a straightforward but limited approach to the CWT that provides a jumping off point for the dual-tree.

2.1 Analyticity vs. finite support

It is often desired in wavelet-based signal processing that the wavelet be well localized in time. (In many applications the wavelet $\psi(t)$ will actually have finite support.) Finitely supported wavelets are of special interest because in this case the discrete wavelet transform (DWT) can be easily implemented with finite impulse response (FIR) filters. However, a finitely supported function can never be exactly analytic, because the Fourier transform of a finitely supported function can never be exactly zero on an interval $[A, B]$ with $B > A$ (on any set of positive measure to be exact) let alone on the entire positive or negative frequency axis [77]. Thus, any exactly analytic wavelet must have infinite support (and slow decay, in fact).

Thus, if we want finitely supported wavelets, then we must accept wavelets that are only *approximately* analytic and a CWT that is only *approximately* magnitude/phase, shift-invariant, and free from aliasing.³ The design challenge will be, of course, to see how close we can get to analyticity. Unfortunately, the standard approach to designing and implementing wavelet transforms (with FIR or IIR filters) has basic limitations even for *approximately* analytic wavelets, as we now illustrate.

2.2 Analyticity vs. perfect reconstruction

The question of how to design filters $h_0(n)$ and $h_1(n)$ satisfying the perfect reconstruction conditions so that the wavelet $\psi(t)$ has short support and vanishing moments was answered by Daubechies (see Sidebar A) [25]. Note, however, that Daubechies' wavelets are not analytic. Can we design the filters $h_i(n)$ in Figure 24 such that the corresponding scaling function and wavelet given by (60) and (59) are complex and (approximately) analytic?

While complex filters satisfying the perfect reconstruction (PR) conditions have been developed [11, 42, 64, 123],

³We can relax the finite support condition, but the resulting infinitely supported wavelets are beyond the scope of this paper.

those solutions do not give analytic wavelets and do not have the desirable properties of analytic wavelets described in the Introduction. (They do, however, have desirable symmetry properties.) It turns out that the design of a complex (approximately) analytic wavelet basis is more difficult than the design of a real wavelet basis. If we follow the standard approach for wavelet design, then problems arise when we require the wavelet to be analytic.

In order that the dyadic dilations and translations of a single function $\psi(t)$ (the wavelet) constitute a basis for signal expansion, $\psi(t)$ must satisfy certain constraints. Unfortunately, these constraints make it difficult to design a wavelet $\psi(t)$ that is also analytic. Specifically, analytic solutions are not possible because the PR conditions (see Sidebar A) require that

$$H_0(e^{j\omega}) \tilde{H}_0(e^{j\omega}) + H_1(e^{j\omega}) \tilde{H}_1(e^{j\omega}) = 2$$

for $-\pi \leq \omega \leq \pi$. Suppose that $h_1(n)$ is (approximately) analytic. Then $H_1(e^{j\omega}) \approx 0$ for $-\pi < \omega < 0$, which in turn implies that $H_0(e^{j\omega}) \tilde{H}_0(e^{j\omega}) \approx 2$ for $-\pi < \omega < 0$. That is, neither $H_0(z)$ nor $\tilde{H}_0(z)$ is a reasonable low-pass filter and consequently the dilation equation does not have a well defined solution. Therefore, the wavelet corresponding to the usual discrete wavelet transform cannot be approximately analytic.

2.3 CWT via DWT post-processing

A natural and straightforward approach towards an invertible analytic CWT splits each output of the FB in Figure 24(a) into its positive and negative frequency components using a complex perfect reconstruction (PR) filter bank acting as a Hilbert transformer [9, 36–39, 108, 109, 114]. But this approach turns out to have a basic limitation.

A complex FB that performs this frequency decomposition can be derived directly from any real 2-channel low-pass/highpass FB with filters $h_0(n)$, $h_1(n)$ by defining the “positive frequency” and “negative frequency” filters as

$$h_p(n) = j^n h_0(n), \quad h_n(n) = j^n h_1(n). \quad (5)$$

This corresponds to a rotation of both filters in the z-plane by 90 degrees. If $h_0(n)$, $h_1(n)$ satisfy the PR conditions, then so will $h_p(n)$, $h_n(n)$. For example, given the low-pass/highpass filters $h_0(n)$, $h_1(n)$ illustrated in the frequency domain in Figure 25, the complex filters $h_p(n)$, $h_n(n)$ are illustrated in the frequency domain in Figure 3. When used by itself, this complex FB can effectively separate the positive and negative frequency components of a signal; in a discrete-time sense, $h_p(n)$ and $h_n(n)$ are approximately analytic.

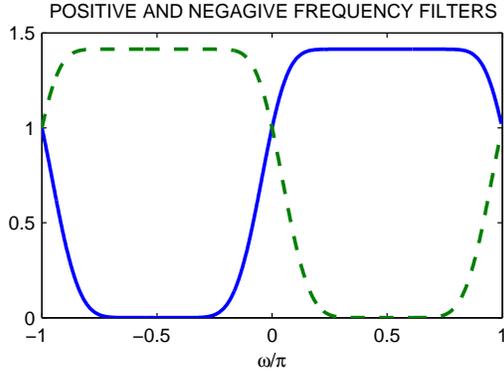


Figure 3: Hilbert transform filter bank (FB). Magnitude frequency responses $|H_p(e^{j\omega})|$ (solid) and $|H_n(e^{j\omega})|$ (dashed) corresponding to (5). $H_p(e^{j\omega})$ approximates $H_a(\omega)$ in (63), while $H_n(e^{j\omega})$ approximates $H_a(-\omega)$.

When this complex FB is used to decompose each subband signal of a real discrete wavelet transform, we obtain the filter bank structure illustrated in Figure 4. Notice that the transform is critically-sampled — the total data rate of the subband signals is equal to the input data rate (although the outputs are now complex).

Although this FB structure is perhaps the most natural approach to developing an approximately analytic discrete wavelet transform, when we examine the overall frequency response of each channel, it becomes apparent that the structure suffers from a basic limitation.

Using z -transforms, consider the filter chain producing the wavelet coefficients at the first level

$$x(n) \longrightarrow \boxed{H_1(z)} \longrightarrow \boxed{\downarrow 2} \longrightarrow \boxed{H_n(z)} \longrightarrow \boxed{\downarrow 2} \longrightarrow c(n).$$

Using the *noble identities* [107], this is equivalent to

$$x(n) \longrightarrow \boxed{H_1(z) H_n(z^2)} \longrightarrow \boxed{\downarrow 4} \longrightarrow c(n).$$

The frequency response of this channel is thus

$$H_{\text{tot}}(z) = H_1(z) H_n(z^2)$$

and in the Fourier domain

$$H_{\text{tot}}(e^{j\omega}) = H_1(e^{j\omega}) H_n(e^{j2\omega}).$$

If $H_1(z)$ and $H_n(z)$ have the frequency responses shown in Figures 25 and 3, then $H_{\text{tot}}(z)$ has the frequency response shown in the second panel of Figure 5.

Observe in Figure 5 that even though the frequency response of each channel is approximately single sided (and thus approximately analytic), there is a substantial bump

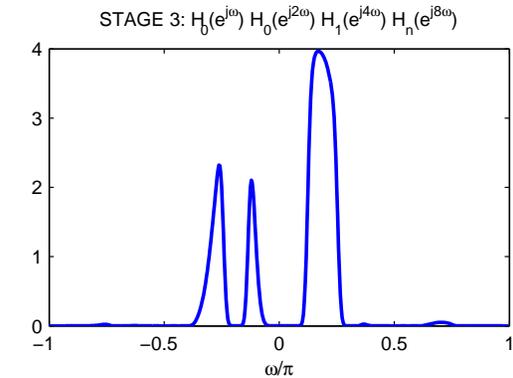
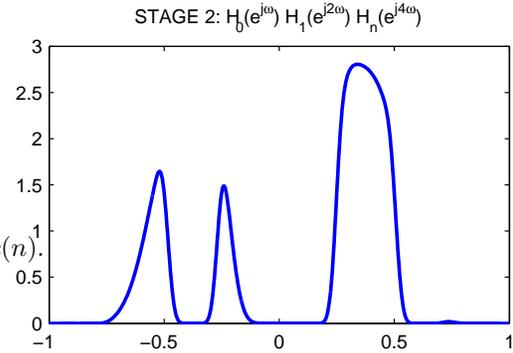
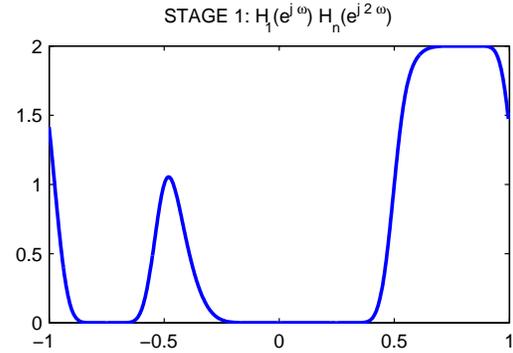
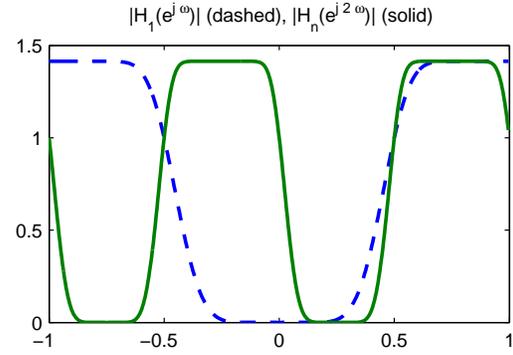


Figure 5: Frequency response for stages 1, 2, and 3 of DWT filter bank with invertible complex post-filtering as in Figure 4.

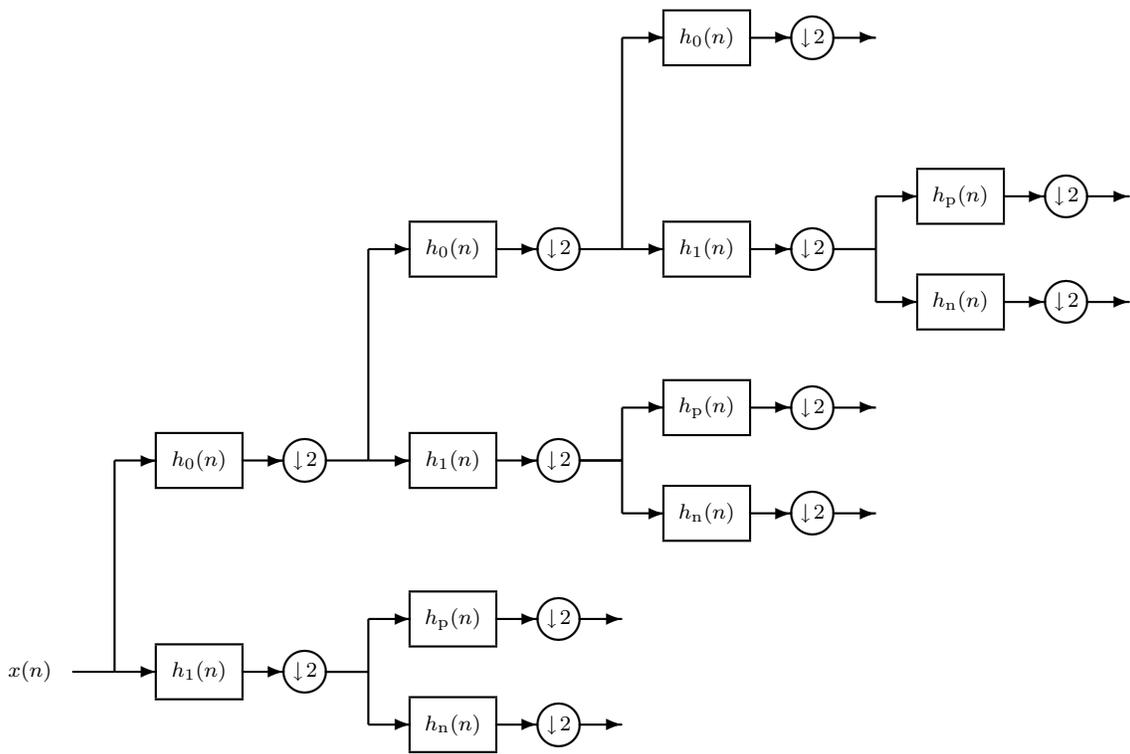


Figure 4: Analysis filter bank for the discrete wavelet transform with invertible complex post-filtering.

on the opposite side of the frequency axis. In fact, this bump is unavoidable for the filter bank structure shown in Figure 4. It is possible to reduce the *width* of the bump by designing $H_1(z)$ and $H_n(z)$ so that they have narrower transition bands, however, then the impulse responses of these filters (and thus the wavelets) will grow longer and they will have a greater degree of ringing. This is contrary to one of the primary goals in wavelet design: short support. Moreover, no matter how long the filters and wavelets are, the *height* of the bump will never diminish. As a consequence of the PR conditions, the bump will always have a height of exactly 1 at $\omega = 0.5\pi$ no matter what filters are used. Figure 5 also illustrates that the problem persists in later FB stages as well.

Even though it has an unavoidable bump on the wrong side of the frequency axis, the CWT generated by the FB in Figure 4 may still be useful for some applications — the frequency response of each channel is largely single sided, the transform is simple to implement, and no new filter design is needed.

However, the *undecimated* discrete wavelet transform can be easily converted into an approximately analytic wavelet transform by using this approach. By decomposing each subband signal of the undecimated DWT with the same complex filter bank considered here, the unwanted bump can be eliminated.⁴ The down-sampling following the real lowpass/highpass filters must be omitted for the bump artifact to be eliminated. (In this case $H_0(z^{2^{(j-1)}})$, $H_1(z^{2^{(j-1)}})$, $H_n(z^{2^{(j-1)}})$, and $H_n(z^{2^{(j-1)}})$ should be used at stage j , for $1 \leq j \leq J$.) Although this approach works with the undecimated DWT, this transform is redundant by a factor of $J + 1$ where J is the number of stages. (An N -point input signal will lead to $(J + 1)N$ wavelet coefficients.) An alternative is the use of the *partially* decimated wavelet transform (PWT) described in [101] to lower the redundancy. The dual-tree CWT, described below, also avoids the unwanted bump and is also expansive, but by just a factor of 2 (for 1-D signals) independent of the number of stages.

2.4 Performing the Hilbert transform first

Another approach to implement an expansive complex wavelet transform first applies a Hilbert transform to the data. The real wavelet transform is then applied to both the original data and the Hilbert transformed data, and the coefficients of each wavelet transform are combined

⁴Note that if the critically-sampled DWT is used and only the down-sampling following the complex positive/negative filters is omitted, then the frequency responses shown in Figure 4 remain unchanged; that is, the bumps will remain.

to obtain a complex wavelet transform [3, 5, 13, 14]. However, note that the ideal Hilbert transform is represented by an infinitely long impulse response that decays very slowly. The use of the ideal (or near ideal) Hilbert transform in conjunction with the wavelet transform effectively increases the support of the wavelets. For the wavelets to have short support, an approximate Hilbert transform more localized in time should be used instead. However, the accuracy of the approximate Hilbert transform should depend on the scale of the wavelet transform (coarse scales should be accompanied by a more accurate Hilbert transform). When the Hilbert transform is applied first to the data, a single Hilbert transform is applied to wavelet coefficients at all scales; and hence it cannot be optimized for all scales simultaneously. On the other hand we shall see that when the Hilbert transform is built into the wavelet transform as in the dual-tree implementation, the Hilbert transform scales with the wavelet scale, as desired.

3 The Dual-Tree Complex Wavelet Transform

As shown in the previous section, the development of an invertible analytic wavelet transform is not as straightforward as might be initially expected. In particular, the filter bank structure illustrated in Figure 24 that is usually used to implement the real discrete wavelet transform does not lend itself to analytic wavelet transforms with desirable characteristics.

3.1 Dual-tree framework

One effective approach for implementing an analytic wavelet transform, first introduced by Kingsbury in 1998, is called the *dual-tree* complex wavelet transform, or dual-tree CWT [54, 55, 57]. Like the idea of positive/negative post-filtering of real subband signals, the idea behind the dual-tree approach is quite simple. The dual-tree CWT employs two *real* DWTs; the first real DWT gives the real part of the transform while the second real DWT gives the imaginary part. The analysis and synthesis filter banks used to implement the dual-tree CWT and its inverse are illustrated in Figures 6 and 7.

The two real wavelet transforms use two different sets of filters, with each satisfying the perfect reconstruction conditions. The two sets of filters are jointly designed so that the overall transform is approximately analytic. Let $h_0(n)$, $h_1(n)$ denote the lowpass/highpass filter pair for the upper filter bank; and let $g_0(n)$, $g_1(n)$ denote

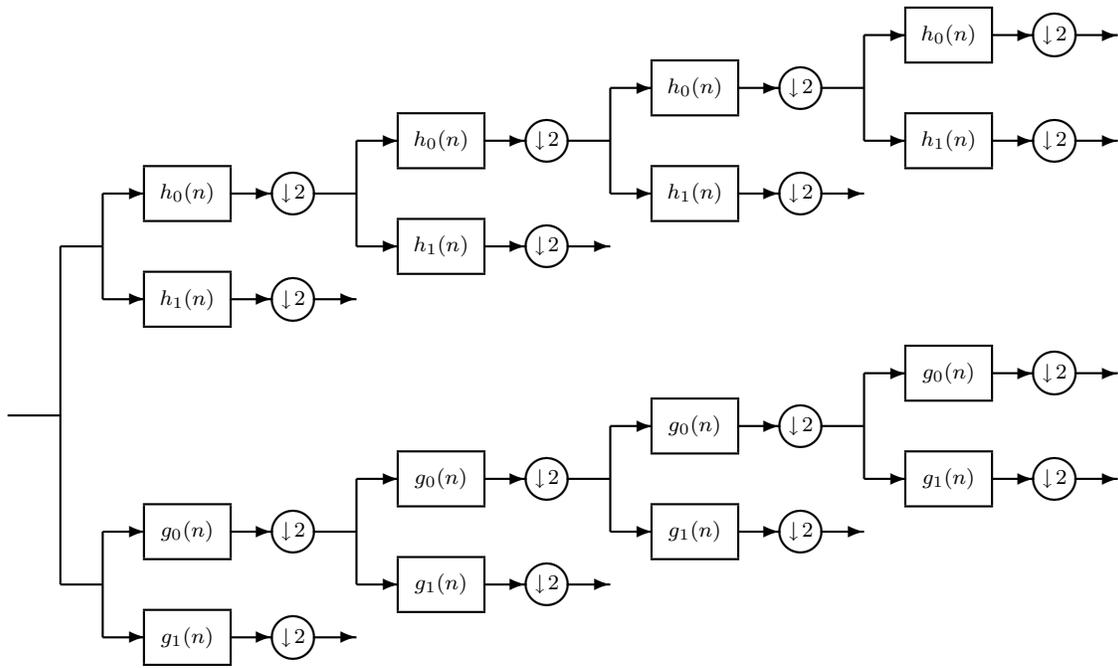


Figure 6: Analysis filter bank for the dual-tree discrete complex wavelet transform (CWT).

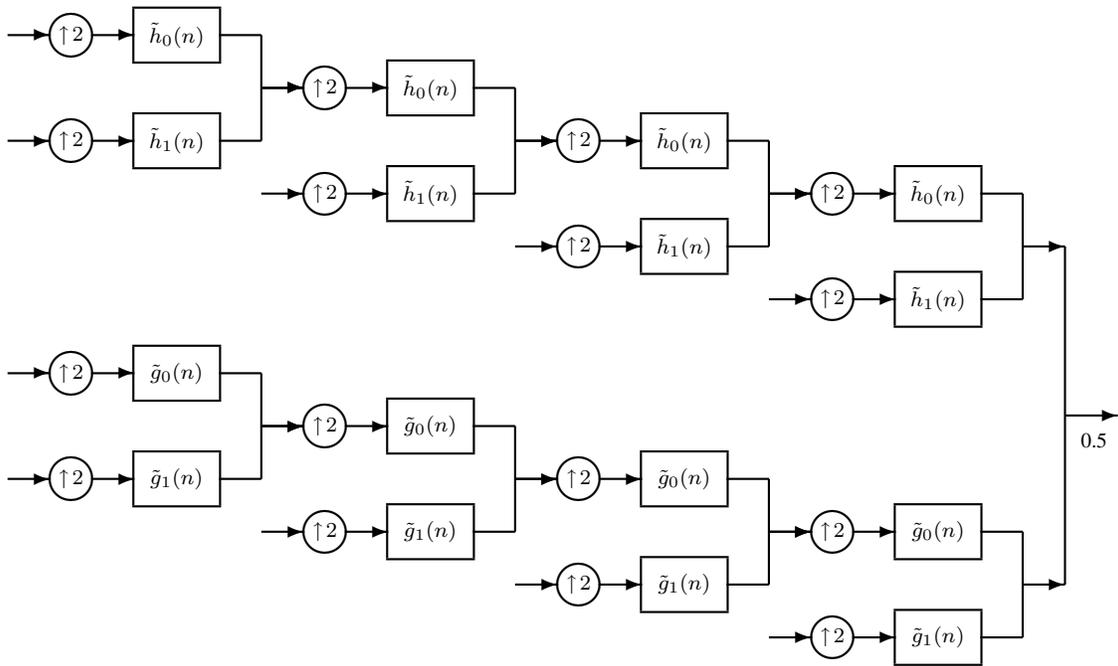


Figure 7: Synthesis filter bank for the dual-tree CWT.

the lowpass/highpass filter pair for the lower filter bank. We will denote the two real wavelets associated with each of the two real wavelet transforms as $\psi_h(t)$ and $\psi_g(t)$. In addition to satisfying the perfect reconstruction conditions, the filters are designed so that the complex wavelet $\psi(t) := \psi_h(t) + j\psi_g(t)$ is approximately analytic. Equivalently, they are designed so that $\psi_g(t)$ is approximately the Hilbert transform of $\psi_h(t)$, [denoted $\psi_g(t) \approx \mathcal{H}\{\psi_h(t)\}$].

Note that the filters are themselves real — no complex arithmetic is required for the implementation of the dual-tree CWT. Also note that the dual-tree CWT is not a critically sampled transform — it is two-times expansive in 1-D because the total output data rate is exactly twice the input data rate.

The inverse of the dual-tree CWT is as simple as the forward transform. To invert the transform, the real part and the imaginary part are each inverted — the inverse of each of the two real DWTs are used — to obtain two real signals. These two real signals are then averaged to obtain the final signal. Note that the original signal $x(n)$ can be recovered from either the real part or the imaginary part alone; however, such inverse dual-tree CWTs do not capture all the advantages an analytic wavelet transform offers.

If the two real DWTs are represented by the square matrices \mathbf{F}_h and \mathbf{F}_g , then the dual-tree CWT can be represented by the rectangular matrix

$$\mathbf{F} = \begin{bmatrix} \mathbf{F}_h \\ \mathbf{F}_g \end{bmatrix}.$$

If the vector \mathbf{x} represents a real signal, then $\mathbf{w}_h = \mathbf{F}_h \mathbf{x}$ represents the real part and $\mathbf{w}_g = \mathbf{F}_g \mathbf{x}$ represents the imaginary part of the dual-tree CWT. The complex coefficients are given by $\mathbf{w}_h + j\mathbf{w}_g$. A (left) inverse of \mathbf{F} is then given by

$$\mathbf{F}^{-1} = \frac{1}{2} \begin{bmatrix} \mathbf{F}_h^{-1} & \mathbf{F}_g^{-1} \end{bmatrix}$$

as we can verify:

$$\mathbf{F}^{-1} \cdot \mathbf{F} = \frac{1}{2} \begin{bmatrix} \mathbf{F}_h^{-1} & \mathbf{F}_g^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{F}_h \\ \mathbf{F}_g \end{bmatrix} = \frac{1}{2} [\mathbf{I} + \mathbf{I}] = \mathbf{I}.$$

We can just as well share the factor of one half between the forward and inverse transforms, to obtain

$$\mathbf{F} := \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{F}_h \\ \mathbf{F}_g \end{bmatrix}, \quad \mathbf{F}^{-1} := \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{F}_h^{-1} & \mathbf{F}_g^{-1} \end{bmatrix}. \quad (6)$$

If the two real DWTs are orthonormal transforms, then the transpose of \mathbf{F}_h is its inverse: $\mathbf{F}_h^t \cdot \mathbf{F}_h = \mathbf{I}$, and similarly for \mathbf{F}_g . In this case, the transpose of the rectangular

matrix \mathbf{F} is also a left inverse: $\mathbf{F}^t \cdot \mathbf{F} = \mathbf{I}$, where we have used (6). That is, the inverse of the dual-tree CWT can be performed using the transpose of the forward dual-tree CWT — it is “self-inverting” in the terminology of [96].

The dual-tree wavelet transform defined in (6) keeps the real and imaginary parts of the complex wavelet coefficients separate. However, the complex coefficients can be explicitly computed using the following form

$$\mathbf{F}_c := \frac{1}{2} \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{I} & -j\mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{F}_h \\ \mathbf{F}_g \end{bmatrix}, \quad (7)$$

$$\mathbf{F}_c^{-1} := \frac{1}{2} \begin{bmatrix} \mathbf{F}_h^{-1} & \mathbf{F}_g^{-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ -j\mathbf{I} & j\mathbf{I} \end{bmatrix}. \quad (8)$$

Note that the complex sum/difference matrix in (7) is unitary (its conjugate transpose is its inverse)

$$\frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & j\mathbf{I} \\ \mathbf{I} & -j\mathbf{I} \end{bmatrix} \cdot \frac{1}{\sqrt{2}} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ -j\mathbf{I} & j\mathbf{I} \end{bmatrix} = \mathbf{I}.$$

(Note that the identity matrix on the RHS is twice the size of those on the LHS.) Therefore, if the two real DWTs are orthonormal transforms then the dual-tree CWT satisfies $\mathbf{F}_c^* \cdot \mathbf{F}_c = \mathbf{I}$, where “*” denotes conjugate transpose. If

$$\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} = \mathbf{F}_c \cdot \mathbf{x}$$

then when \mathbf{x} is real we have $\mathbf{v} = \mathbf{u}^*$ so \mathbf{v} need not be computed. When the input signal \mathbf{x} is complex, then $\mathbf{v} \neq \mathbf{u}^*$ so both \mathbf{u} and \mathbf{v} need to be computed.

When the dual-tree CWT is applied to a real signal, the output of the upper and lower filter banks in Figure 6 will be the real and imaginary parts of the complex coefficients, and they can be stored separately, as represented by (6). However, if the dual-tree CWT is applied to a complex signal, then the output of both the upper and lower filter banks will be complex, and it is no longer correct to label them as the real and imaginary parts. For complex input signals, the form in (7) is more appropriate. For a real N -point signal, the form in (7) yields $2N$ complex coefficients, but N of these coefficients are the complex conjugates of the other N coefficients. For a general complex N -point signal, the form in (7) yields $2N$ general complex coefficients. Therefore, for both real and complex input signals, the CWT is two-times expansive.

When the two real DWTs are orthonormal and the $1/\sqrt{2}$ factor is included as in (6), the dual-tree CWT gains a Parseval’s energy theorem: the energy of the input signal is equal to the energy in the wavelet domain

$$\sum_{j,n} (|d_h(j,n)|^2 + |d_g(j,n)|^2) = \sum_n |x(n)|^2.$$

The dual-tree CWT is also easy to implement. Because there is no data flow between the two real DWTs, they can each be implemented using existing DWT software and hardware. Moreover, the transform is naturally parallelized for efficient hardware implementation. In addition, because the dual-tree CWT is implemented using two real wavelet transforms, the use of the dual-tree CWT can be informed by the existing theory and practice of real wavelet transforms. For example, criteria for wavelet design (vanishing moments, etc) and wavelet-based signal processing algorithms (thresholding of wavelet coefficients, and so on) that have been developed for real wavelet transforms can also be applied to the dual-tree CWT.

It should be noted, however, that the dual-tree CWT requires the design of new filters. Primarily, the dual-tree CWT requires a *pair* of filter sets chosen so that the corresponding wavelets form an approximate Hilbert transform pair. Existing filters for wavelet transforms should not be used to implement both trees of the dual-tree CWT. For example, pairs of Daubechies' wavelet filters do not satisfy the requirement that $\psi_g(t) \approx \mathcal{H}\{\psi_h(t)\}$. If the dual-tree wavelet transform is implemented with filters not satisfying this requirement, then the transform will not provide the full advantages of analytic wavelets described in the Introduction.

3.2 The half-sample delay condition

Translating wavelet properties into filter properties translates the wavelet design problem into a filter design problem. For example, it is well known that a wavelet $\psi(t)$ has K vanishing moments if the transfer function of the lowpass filter has the form $H_0(z) = (1+z)^K Q(z)$ for some $Q(z)$.

The dual-tree CWT inspires a new filter design problem: what property should the two lowpass filters $h_0(n)$ and $g_0(n)$ satisfy so as to ensure that the corresponding wavelets form an approximate Hilbert transform pair, that is $\psi_g(t) \approx \mathcal{H}\{\psi_h(t)\}$? Here

$$\psi_h(t) = \sqrt{2} \sum_n h_1(n) \phi_h(t),$$

$$\phi_h(t) = \sqrt{2} \sum_n h_0(n) \phi_h(t),$$

$h_1(n) = (-1)^n h_0(d-n)$; $\psi_g(t)$, $\phi_g(t)$, and $g_1(n)$ are defined similarly.⁵ Since the wavelets depend on the scaling functions, and since the scaling functions depend on

⁵For convenience, we assume here that the wavelet transform is orthonormal.

the filters only implicitly, it is not at first obvious how the filters should be related. However, it turns out that the two lowpass filters should satisfy a very simple property: *one of them should be approximately a half-sample shift of the other* [87]

$$g_0(n) \approx h_0(n - 0.5) \implies \psi_g(t) \approx \mathcal{H}\{\psi_h(t)\}. \quad (9)$$

Since $g_0(n)$ and $h_0(n)$ are defined only on the integers, this statement is somewhat informal. However, we can make the statement rigorous using Fourier transforms. In [87] it is shown that if $G_0(e^{j\omega}) = e^{-j0.5\omega} H_0(e^{j\omega})$ then $\psi_g(t) = \mathcal{H}\{\psi_h(t)\}$. The converse has been proved in [76, 122], making the condition necessary and sufficient. The necessary and sufficient conditions for the biorthogonal case were proved in [121]. To understand intuitively why the half-sample delay condition leads to a nearly shift-invariant discrete wavelet transform, note that the half-sample delay condition is equivalent to uniformly oversampling the lowpass signal at each scale by 2:1, thus largely avoiding the aliasing due to the lowpass downsamplers [53–55].

It will be useful to rewrite the half-sample delay condition in terms of the magnitude and phase functions separately:

$$|G_0(e^{j\omega})| = |H_0(e^{j\omega})|, \quad (10)$$

$$\angle G_0(e^{j\omega}) = \angle H_0(e^{j\omega}) - 0.5\omega. \quad (11)$$

Equivalently, $g_0(n)$ could be obtained from $h_0(n)$ by filtering $h_0(n)$ with an ideal fractional delay system. However, such a system is not realizable — its impulse response is of infinite length and its transfer function is not rational. Even if it were realizable it might not give a desirable solution because if $h_0(n)$ is FIR, then $g_0(n)$ would be of infinite length. Indeed, if $\psi_h(t)$ is a wavelet of finite support, then its exact Hilbert transform will have infinite support. Therefore, in practical implementations of the dual-tree CWT, the delay condition (10) and (11) will be satisfied only approximately; the wavelets $\psi_h(t)$ and $\psi_g(t)$ will form only an approximate Hilbert pair; and the complex wavelet $\psi_h(t) + j\psi_g(t)$ will be only approximately analytic.

A question remains, however: is it possible to satisfy simultaneously the perfect reconstruction condition (55) exactly and the half-sample delay condition (10), (11) approximately with *short* filters? Or does the dual-tree CWT have some side effect that limits its effectiveness as an analytic wavelet transform (like the bumps in Figure 5) when short filters are used? The next section describes several methods for filter design for the dual-tree CWT which demonstrates that with relatively short filters an effective

invertible approximately analytic wavelet transform can indeed be implemented using the dual-tree approach.

3.3 Filter design for the dual-tree CWT

As in the case of filter design for real wavelet transforms, there are various approaches to the design of filters for the dual-tree CWT. In the following, we describe methods to construct filters satisfying the following desired properties:

1. Approximate half-sample delay property
2. Perfect reconstruction (orthogonal or biorthogonal)
3. Finite support (FIR filters)
4. Vanishing moments/good stopband
5. Linear-phase filters (desired, but not required of a wavelet transform for it to be approximately analytic). Moreover, only the *complex* filter responses need be linear-phase; this can be achieved by taking $g_0(n) = h_0(N - 1 - n)$.

One approach to dual-tree filter design is to let $h_0(n)$ be some existing wavelet filter. Then, given $h_0(n)$, we need to design $g_0(n)$ so as to simultaneously satisfy (i) $G_0(e^{j\omega}) \approx e^{-j0.5\omega} H_0(e^{j\omega})$ and (ii) the perfect reconstruction conditions. (Algorithms for designing an orthonormal wavelet basis to match a specified signal class are described, for example, in [20].) Unfortunately, this will sometimes result in $g_0(n)$ being substantially longer than $h_0(n)$ (but see [105, 121]). By jointly designing $h_0(n)$ and $g_0(n)$, we can obtain a pair of filters of equal (or near-equal) length, where both are relatively short. It should be noted however, that filters for the dual-tree CWT are generally somewhat longer than filters for real wavelet transforms with similar numbers of vanishing moments, because of the additional constraints (10)-(11) the filters must approximately satisfy.

In the following, we describe three methods for FIR dual-tree filter design. Fast implementations of some of these filters have been recently described in [1].

3.3.1 Linear-phase biorthogonal solution

The first solution, introduced in [53, 54], sets $h_0(n)$ to be a symmetric odd-length (Type I) FIR filter and sets $g_0(n)$ to be a symmetric even-length (Type II) FIR filter, such that for N odd:

$$h_0(n) = h_0(N - 1 - n), \quad (12)$$

$$g_0(n) = g_0(N - n). \quad (13)$$

This solution must be a biorthogonal solution (the filters in the synthesis filter bank are not time-reversed versions of the filters in the analysis filter bank). This is because real orthonormal FIR two-channel filter banks cannot be symmetric (except for the Haar solution). Note that if $h_0(n)$ is a symmetric N -point impulse response (supported on $0 \leq n \leq N - 1$) then $\angle H_0(e^{j\omega}) = -0.5(N - 1)\omega$. Similarly, if $g_0(n)$ is a symmetric $(N + 1)$ -point impulse response (supported on $0 \leq n \leq N$) then $\angle G_0(e^{j\omega}) = -0.5N\omega$. Therefore, for this type of solution, the phase part (11) of the half-sample delay condition is exactly satisfied, but the magnitude part (10) is not:

$$|G_0(e^{j\omega})| \neq |H_0(e^{j\omega})|, \quad (14)$$

$$\angle G_0(e^{j\omega}) = \angle H_0(e^{j\omega}) - 0.5\omega. \quad (15)$$

Therefore, $h_0(n)$ and $g_0(n)$ should be design so as to approximately satisfy the magnitude condition (10).

The design of a pair of symmetric perfect reconstruction (biorthogonal) filters approximately satisfying the magnitude relation (10) is performed in [53, 54] by an iterative error minimization strategy rather similar to that in [58]. Alternative techniques are given in [105] which employ even-length Bernstein filter banks (EBFBs) to obtain the matching even length filters.

3.3.2 q-shift solution

The second solution, introduced in [56], sets

$$g_0(n) = h_0(N - 1 - n) \quad (16)$$

where N , now even, is the length of $h_0(n)$, which is supported on $0 \leq n \leq N - 1$. In this case, the magnitude part (10) of the half-sample delay condition is exactly satisfied due to the time-reverse relation between the filters, but the phase part (11) is not exact:

$$|G_0(e^{j\omega})| = |H_0(e^{j\omega})|, \quad (17)$$

$$\angle G_0(e^{j\omega}) \neq \angle H_0(e^{j\omega}) - 0.5\omega. \quad (18)$$

Thus the filters must be designed so that the phase condition is approximately satisfied.

The q-shift solution has an interesting property that leads to its name: If you ask that $g_0(n)$ and $h_0(n)$ be related as in (16) and also that they approximately satisfy (11), then it turns out that the frequency response of $h_0(n)$ has approximately linear phase. This is verified by writing (16) in terms of Fourier transforms:

$$G_0(e^{j\omega}) = \overline{H_0(e^{j\omega})} e^{-j(N-1)\omega}$$

where the overbar represents complex conjugation. This implies that the phases satisfy

$$\angle G_0(e^{j\omega}) = -\angle H_0(e^{j\omega}) - (N-1)\omega.$$

If the two filters satisfy the phase condition (11) approximately (that is, $\angle G_0(e^{j\omega}) \approx \angle H_0(e^{j\omega}) - 0.5\omega$) then

$$\angle H_0(e^{j\omega}) - 0.5\omega \approx -\angle H_0(e^{j\omega}) - (N-1)\omega$$

from which we have

$$\angle H_0(e^{j\omega}) \approx -0.5(N-1)\omega + 0.25\omega. \quad (19)$$

That is, $h_0(n)$ is an approximately linear-phase filter. This also says that $h_0(n)$ is approximately symmetric around the point $n = 0.5(N-1) - 0.25$. Note that this is one quarter away from the “natural” point of symmetry (if $h_0(n)$ were exactly symmetric), and for this reason solutions of this kind were introduced as *quarter-shift* (q-shift) dual-tree filters in [56].

For the q-shift solution, the wavelets are related by

$$\psi_g(t) = \psi_h(N-1-t).$$

The imaginary part of the complex wavelet is a time-reversed version of the real part. Therefore the q-shift solution produces complex wavelets that are exactly linear-phase (regardless of what filters $h_0(n)$, $g_0(n)$ are used).

The q-shift solution calls for the design of a single filter satisfying simultaneously the perfect reconstruction conditions and the phase condition (19); and true orthonormal solutions are possible here, because the filters need only be *approximately* linear phase and their coefficients do not need to exhibit symmetry. The same time-reverse condition then applies between analysis and synthesis filters as between the dual trees, yielding a surprisingly neat overall solution from a single filter design. In [56], orthonormal solutions to this design problem are found by optimization over lattice angles, using a lattice parameterization of orthonormal filter banks. One of these q-shift filters has only six non-zero coefficients, making it efficient for implementation. Longer filters have been obtained using an iterative frequency domain error minimization criterion [58], which is better suited to the design of longer q-shift filters (typically using 12 or more taps) with improved smoothness and shift-invariance properties.

3.3.3 Common-factor solution

The third solution, introduced in [88], can be used to design both orthonormal and biorthogonal solutions for the dual-tree CWT. In this approach we set

$$h_0(n) = f(n) * d(n), \quad (20)$$

$$g_0(n) = f(n) * d(L-n) \quad (21)$$

where $*$ represents discrete-time convolution and where $d(n)$ is supported on $0 \leq n \leq L$. Equivalently

$$H_0(z) = F(z) D(z), \quad (22)$$

$$G_0(z) = F(z) z^{-L} D(1/z). \quad (23)$$

Like the q-shift solution, for solutions of this kind the magnitude part (10) of the half-sample delay condition is exactly satisfied but the phase part (11) is not:

$$|G_0(e^{j\omega})| = |H_0(e^{j\omega})|, \quad (24)$$

$$\angle G_0(e^{j\omega}) \neq \angle H_0(e^{j\omega}) - 0.5\omega. \quad (25)$$

The filters must be designed so that the phase condition is approximately satisfied. From (22)-(23) we have

$$G_0(z) = H_0(z) A(z) \quad (26)$$

where

$$A(z) := \frac{z^{-L} D(1/z)}{D(z)}$$

is an allpass transfer function — it has the property that $|A(e^{j\omega})| = 1$. Therefore, from (26), $|G_0(e^{j\omega})| = |H_0(e^{j\omega})|$ and

$$\angle G_0(e^{j\omega}) = \angle H_0(e^{j\omega}) + \angle A(e^{j\omega}).$$

If the filters $h_0(n)$ and $g_0(n)$ are to satisfy the phase condition (11) approximately, then $D(z)$ must be chosen so that

$$\angle A(e^{j\omega}) \approx -0.5\omega. \quad (27)$$

With (27) we find that $A(z)$ should be a fractional delay allpass system.

A solution to the dual-tree filter design problem where the filters are taken to have the form in (20)-(21), can be found in two steps: First, find an FIR $D(z)$ so that $A(z)$ satisfies (27). Second, find an FIR $F(z)$ so that $h_0(n)$ and $g_0(n)$ satisfy the perfect reconstruction conditions.

The first step can draw on existing literature. The design of allpass systems with phase response (27) is already well studied [61, 62, 85]. The formula for the maximally flat-delay all-pass filter, adapted from Thiran’s filter in [106], is

$$D(z) = 1 + \sum_{n=1}^L \binom{L}{n} \left[\prod_{k=0}^{n-1} \frac{\tau - L + k}{\tau + 1 + k} \right] (-z)^{-n}. \quad (28)$$

With this $D(z)$, we have $A(e^{j\omega}) \approx e^{-j\tau\omega}$ around $\omega = 0$. We can use $D(z)$ in (28) with $\tau = 0.5$. The phase of the maximally flat fractional-delay all-pass system $A(z)$ is illustrated in Figure 8 for $L = 1, 2, 3$. For larger values of L an improved approximation to 0.5ω is obtained. The

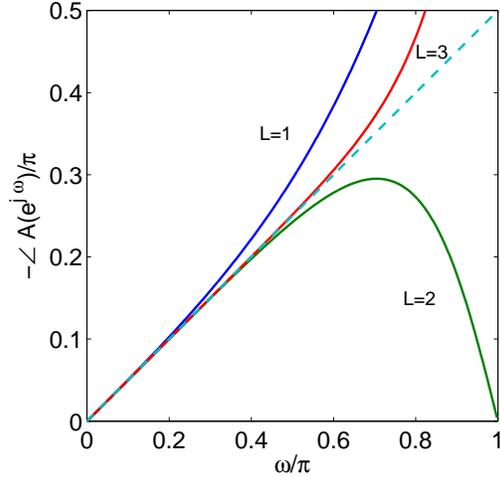


Figure 8: The phase $\angle A(e^{j\omega})$ of the maximally flat fractional-delay all-pass system with $\tau = 0.5$ and $L = 1, 2, 3$.

line 0.5ω is indicated in the figure by the dashed line. Note that the behavior of the phase in the stopband of the lowpass filter $H_0(z)$ is not important, so the deviation of the phase from 0.5ω near $\omega = \pi$ is not relevant. Other fractional delay allpass filters can also be used; in [38] a different allpass filter is used.

The second step, finding $F(z)$ so that $h_0(n)$ and $g_0(n)$ satisfy the PR conditions, requires only a solution to a linear system of equations and a spectral factorization. As described in [88] this design procedure allows for an arbitrary number of vanishing wavelet moments to be specified.

This approach to the dual-tree filter design problem is exactly analogous to Daubechies' construction of short orthonormal (and biorthogonal) wavelet bases with vanishing moments. Like the Daubechies' construction, if the common-factor approach is used to design an orthonormal wavelet transform, then the filters will not be symmetric. However, also similar to the Daubechies' construction, if this approach is used to design a biorthogonal transform, then the filter $f(n)$ can be exactly symmetric and the filters $h_0(n)$ and $g_0(n)$ will be approximately linear-phase (because $d(n)$ has approximately linear phase).

3.3.4 Examples

A q -shift Hilbert pair of wavelets is illustrated in Figure 9. The filters were obtained using the design algorithm in [58] and are of length 14. The spectrum of the complex wavelet $\psi_h(t) + j\psi_g(t)$ is shown in the figure, and it is clearly nearly analytic (approximately zero on the negative frequency axis). A common factor Hilbert pair of

wavelets based on a biorthogonal set of filters is illustrated in Figure 10. The filters were obtained using the design algorithm in [88] and have 2 vanishing moments each. The analysis lowpass filters are of length 11 and the synthesis lowpass filters are of length 13.

3.4 Implementation issues

It turns out that the implementation of the dual-tree CWT requires that the first stage of the dual-tree filter bank be different from the succeeding stages. If the same perfect reconstruction filters are used for each stage, as Figure 6 indicates, then the first several stages of the filter bank will not be approximately analytic; that is, the frequency responses for these stages will not be approximately single-sided. In this section, we describe how the filters for the first stage should be chosen so that the dual-tree CWT is approximately analytic for every stage.

Note that the half-sample delay condition, $g_0(n) \approx h_0(n - 0.5)$, was derived by asking that $\psi_g(t) \approx \mathcal{H}\{\psi_h(t)\}$. However, $\psi_g(t)$ and $\psi_h(t)$ are defined on the real line through Equations (59), (60), and they do not always accurately reflect the behavior and properties of the filter bank for the first several stages. These functions are most useful for understanding the behavior of the filter bank at stage j as $j \rightarrow \infty$.

To understand how the filters at each stage of the dual-tree filter bank should be designed, it is useful to consider again the half-sample delay condition. It turns out that if the lowpass filters satisfy the half-sample delay condition, $g_0(n) \approx h_0(n - 0.5)$, then the scaling functions also satisfy a half-sample delay condition: $\phi_g(t) \approx \phi_h(t - 0.5)$. The wavelet expansion of a signal $x(t)$ on the real line in (1) calls for the integer translates of the scaling function $\phi(t)$. Therefore, the condition $\phi_g(t) \approx \phi_h(t - 0.5)$ implies that the integer translates of $\phi_g(t)$ fall midway between the integer translates of $\phi_h(t)$. That is, the two scaling functions satisfy an *interlacing* property. For the discrete form of the dual-tree CWT to be (approximately) analytic at each stage j , it is necessary that the dual-tree filter bank duplicate this interlacing property.

Instead of using the same filters at each stage of the dual-tree filter bank, as depicted in Figure 6, let us suppose that at each stage we use a different set of perfect reconstruction filters. As illustrated in Figure 11, the lowpass filters used at stage j will be denoted by $h_0^{(j)}(n)$ and $g_0^{(j)}(n)$. (At each stage, in each tree, the highpass filter will be determined by the lowpass filter, as usual.)

From the input of the filter bank to the lowpass output of the upper filter bank at stage j we have (by basic mul-

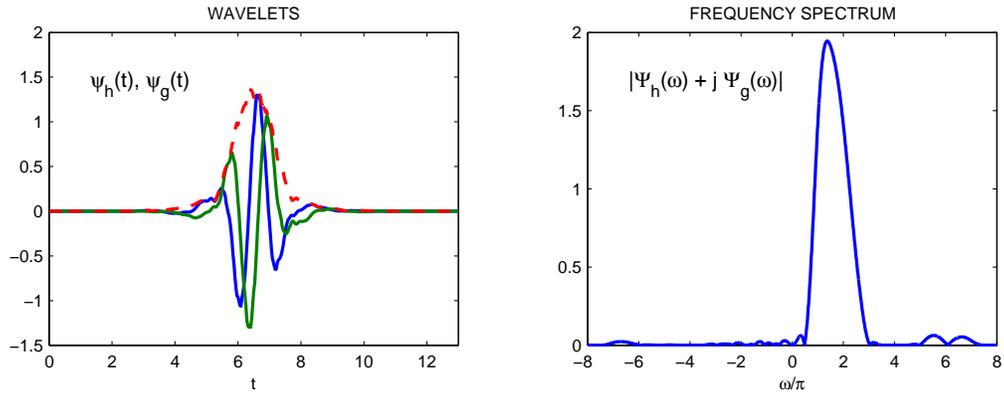


Figure 9: q -shift complex wavelet corresponding to a set of orthonormal dual-tree filters of length 14 [58].

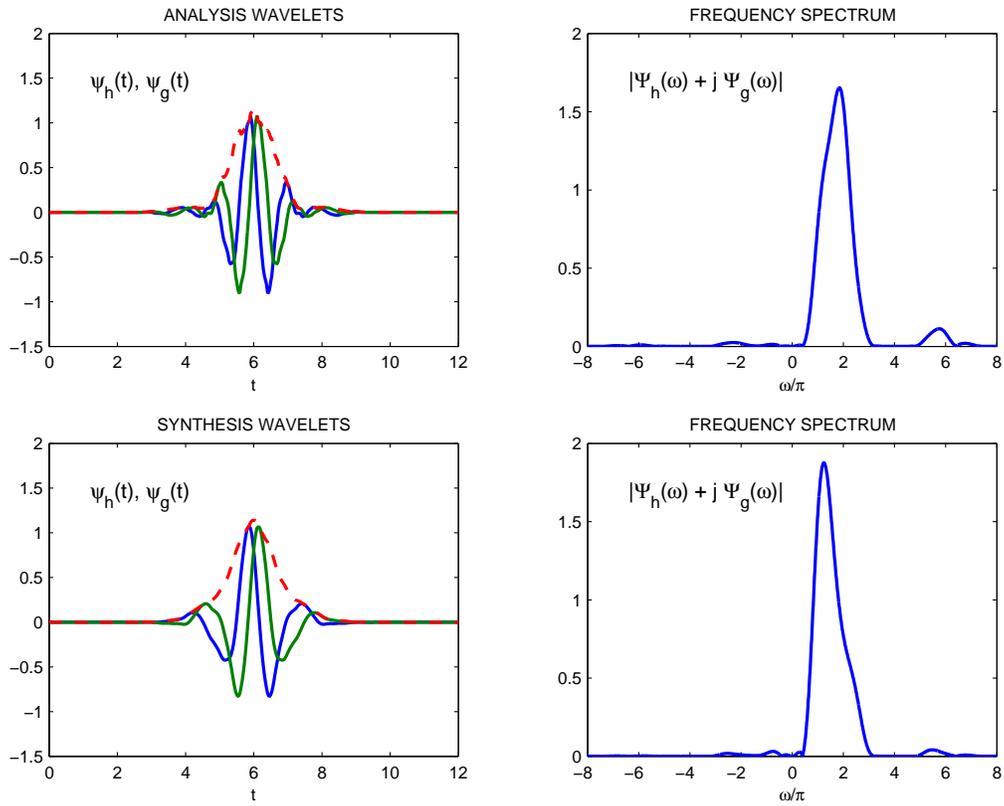


Figure 10: Common factor complex wavelet corresponding to a set of biorthogonal dual-tree filters [88].

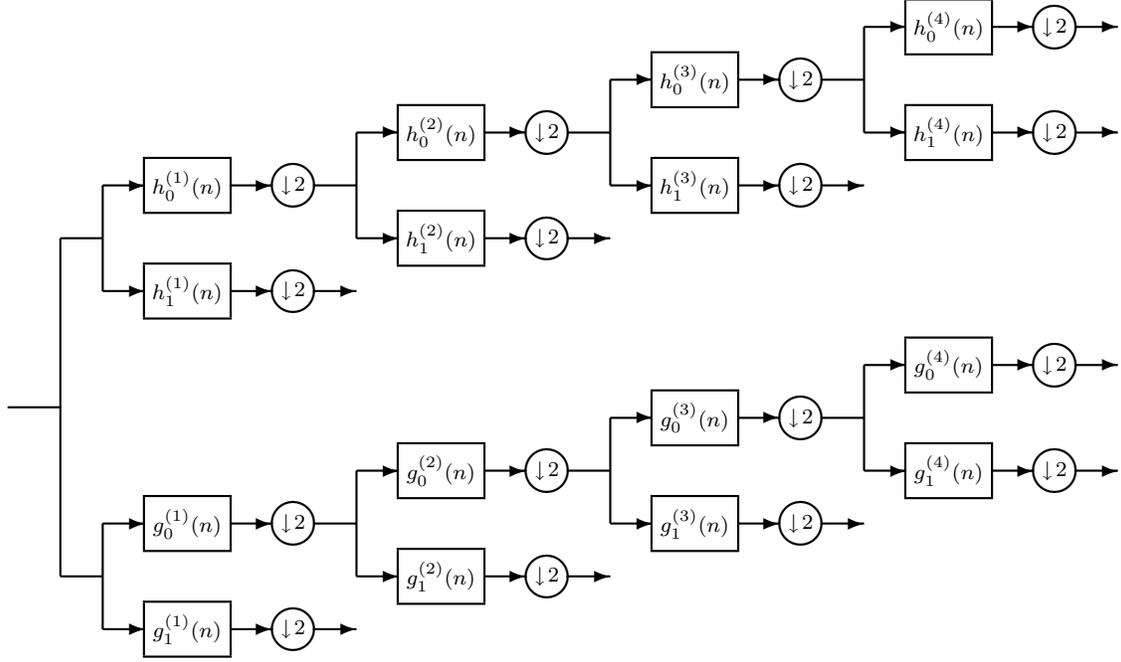


Figure 11: Analysis filter bank for the dual-tree CWT with a different set of filters at each stage.

tirate properties) the system

$$x(n) \longrightarrow \boxed{h_{\text{tot}}^{(j)}(n)} \longrightarrow \boxed{\downarrow 2^j} \longrightarrow$$

where $h_{\text{tot}}^{(j)}(n)$ is given by

$$H_{\text{tot}}^{(j)}(z) = H_0^{(1)}(z) H_0^{(2)}(z^2) \cdots H_0^{(j)}(z^{2^{j-1}}). \quad (29)$$

We have similar expression for $G_{\text{tot}}^{(j)}(z)$ in the lower filter bank.

To ensure that the discrete analysis functions of the dual-tree CWT satisfy the interlacing property, we require that the filters at each stage, $h_0^{(j)}(n)$ and $g_0^{(j)}(n)$, be designed so that the translates of $g_{\text{tot}}^{(j)}(n)$ by 2^j fall midway between the translates of $h_{\text{tot}}^{(j)}(n)$ by 2^j . At stage 1 for example, we require that the translates of $g_{\text{tot}}^{(1)}(n)$ by 2 fall midway between the translates of $h_{\text{tot}}^{(1)}(n)$ by 2. That is, we require that

$$g_{\text{tot}}^{(1)}(n) \approx h_{\text{tot}}^{(1)}(n-1).$$

At stage 2, we require that the translates of $g_{\text{tot}}^{(2)}(n)$ by 4 fall midway between the translates of $h_{\text{tot}}^{(2)}(n)$ by 4. That is, we require that

$$g_{\text{tot}}^{(2)}(n) \approx h_{\text{tot}}^{(2)}(n-2).$$

At stage 3, we require that

$$g_{\text{tot}}^{(3)}(n) \approx h_{\text{tot}}^{(3)}(n-4),$$

and so forth.

At stage $j = 1$, $h_{\text{tot}}^{(1)}(n)$ is just $h_0^{(1)}(n)$, and we are asking that

$$g_0^{(1)}(n) \approx h_0^{(1)}(n-1). \quad (30)$$

This is different (and easier!) from the half-sample delay condition discussed above. Dual-tree filters designed so as to satisfy the half-sample delay condition should not be used for the first stage. For the first stage, the condition (30) can be satisfied exactly by using the same set of filters in each of the two trees; it is necessary only to translate one set of filters by one sample with respect to the other set. Moreover, any set of perfect reconstruction filters can be used for the first stage.

For stages $j > 1$ it is more useful to write the requirements using the frequency responses of the filters. For stage $j = 2$, we require that

$$G_{\text{tot}}^{(2)}(e^{j\omega}) \approx e^{-j2\omega} H_{\text{tot}}^{(2)}(e^{j\omega}). \quad (31)$$

Using (29) we can write (31) in terms of the individual filters as

$$G_0^{(1)}(e^{j\omega}) G_0^{(2)}(e^{j2\omega}) \approx e^{-j2\omega} H_0^{(1)}(e^{j\omega}) H_0^{(2)}(e^{j2\omega}). \quad (32)$$

We already have $G_0^{(1)}(e^{j\omega}) \approx e^{-j\omega} H_0^{(1)}(e^{j\omega})$ from (30), and so from (32) we obtain

$$G_0^{(2)}(e^{j2\omega}) \approx e^{-j\omega} H_0^{(2)}(e^{j2\omega})$$

or equivalently

$$G_0^{(2)}(e^{j\omega}) \approx e^{-j0.5\omega} H_0^{(2)}(e^{j\omega}) \quad (33)$$

or $g_0^{(2)}(n) \approx h_0^{(2)}(n - 0.5)$. This is the half-sample delay condition we have already encountered.

For stage $j = 3$, we require that

$$G_{\text{tot}}^{(3)}(e^{j\omega}) \approx e^{-j4\omega} H_{\text{tot}}^{(3)}(e^{j\omega}). \quad (34)$$

Using (29) we can write (34) in terms of the individual filters as

$$G_0^{(1)}(e^{j\omega}) G_0^{(2)}(e^{j2\omega}) G_0^{(3)}(e^{j4\omega}) \approx e^{-j4\omega} H_0^{(1)}(e^{j\omega}) H_0^{(2)}(e^{j2\omega}) H_0^{(3)}(e^{j4\omega}). \quad (35)$$

We already have $G_0^{(1)}(e^{j\omega}) \approx e^{-j\omega} H_0^{(1)}(e^{j\omega})$ from (30) and $G_0^{(2)}(e^{j\omega}) \approx e^{-j0.5\omega} H_0^{(2)}(e^{j\omega})$ from (33), and so from (35) we obtain

$$G_0^{(3)}(e^{j4\omega}) \approx e^{-j2\omega} H_0^{(3)}(e^{j4\omega})$$

or equivalently

$$G_0^{(3)}(e^{j\omega}) \approx e^{-j0.5\omega} H_0^{(3)}(e^{j\omega})$$

or $g_0^{(3)}(n) \approx h_0^{(3)}(n - 0.5)$. This is once again the half-sample delay condition.

Using the same derivation for further stages, it turns out that for each stage, $j > 1$, we always obtain the same condition

$$g_0^{(j)}(n) \approx h_0^{(j)}(n - 0.5).$$

Therefore, the perfect reconstruction dual-tree filters introduced previously can be used for each stage of the dual-tree filter bank after the first stage. Only the first stage requires a different set of filters. Moreover, any existing PR filters can be used for the first stage — it is only required to offset them from each other by one sample.

Since the first-stage filters do not need to satisfy approximately the conditions (10)-(11), they can be the same length as those used for a real wavelet transform (the filters for the following stages will be somewhat longer). For a 2-D wavelet transform, these filters consume about 3/4 of the total execution time, and so their length can be important for implementation efficiency.

Figure 12 illustrates the frequency responses of stages 1 through 4 of the dual-tree CWT. The first stage is

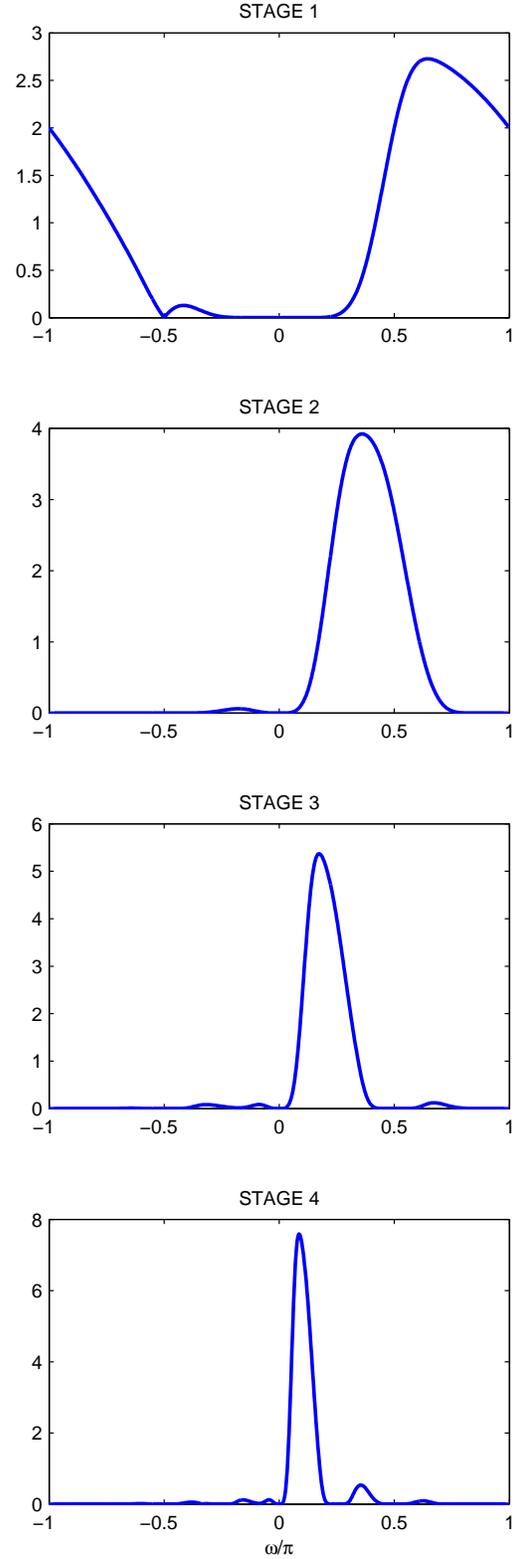


Figure 12: Frequency responses of the (approximately analytic) dual-tree CWT for stages 1 through 4. Compare with Figure 5.

quite far from being analytic, however, the later stages are quite close to being analytic. For every stage after the first stage, the frequency responses of the complex filters are close to being single-sided and are free of the unwanted lobes on the opposite side of the frequency axis that are present in Figure 5. In this example, $h_0^{(1)}(n)$ is a Daubechies length-10 filter, $g_0^{(1)}(n) = h_0^{(1)}(n - 1)$, and $g_i(n)$, $h_i(n)$ are orthonormal solutions of length 12 designed according to the algorithm of Section 3.3.3.

3.4.1 Swapping

We saw above that the filters for the first dual-tree stage should be different from the filters for the remaining stages. There is another implementation detail. It was suggested in [55] that for each stage $j > 2$ the filters should be interchanged in the upper and lower filter banks. That is, the upper filter bank should use the filters $h_0(n)$ and $h_1(n)$ for the even stages $j = 2, 4, 6, \dots$ and the filters $g_0(n)$ and $g_1(n)$ for the odd stages $j = 3, 5, 7, \dots$. Correspondingly, the filters in the lower filter bank should also alternate. This scheme is illustrated in Figure 13. By alternating filters from stage to stage (except the first stage), in the cases when $|G_0(e^{j\omega})| \neq |H_0(e^{j\omega})|$, a more balanced implementation is obtained. (The delay differences must *not* be swapped, even when the filters are swapped, so an extra delay of one sample must be included as required to keep the polarity of the half-sample delay correct at each level.)

We note, however, that use of alternating filters is not required to achieve analytic behavior in the complex filters. Hence, this implementation detail is less important than using a different filter set for the first stage.

4 2-D Dual-Tree Complex Wavelet Transform

4.1 Oriented wavelets

The multi-dimensional (M-D) dual-tree CWT both maintains the attractive properties of the 1-D dual-tree and gains additional properties that make it particularly effective for M-D wavelet-based signal processing. In particular, M-D dual-tree wavelets are not only approximately analytic but also *oriented* and thus natural for analyzing and processing oriented singularities like edges in images and surfaces in 3-D datasets.

Although wavelet bases are optimal in a sense for a large class of 1-D signals, the 2-D wavelet transform does not possess these optimality properties for natural images

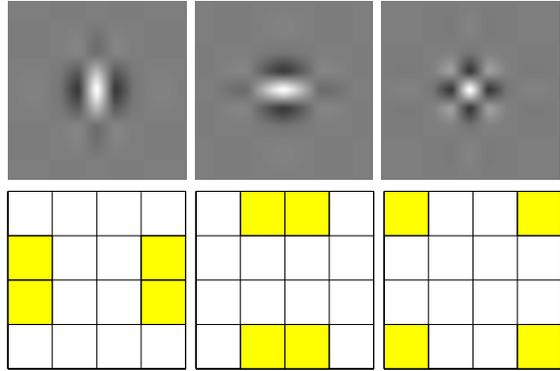


Figure 14: Typical wavelets associated with the 2-D separable DWT. Top row illustrates the wavelets in the space domain (LH, HL, HH); bottom row illustrates the (idealized) support of the Fourier spectrum of each wavelet in the 2-D frequency domain (the origin lies at the center). The checkerboard artifact of the third wavelet is evident.

[33, 112]. The reason for this is that while the separable 2-D wavelet transform represents point-singularities efficiently, it is less efficient for line- and curve-singularities (edges). Thus, one of the interesting avenues in wavelet-related research has been the development of 2-D multi-scale transforms that represent edges more efficiently than the separable DWT. Examples include steerable pyramids [41, 96], directional filter banks and pyramids [10, 31], curvelets [15, 100], and directional wavelet transforms based on complex filter banks [36, 39, 55, 57]. These transforms isolate edges with different orientations in different subbands, and they frequently give superior results in image processing applications compared to the separable DWT.

The separable (row-column) implementation of the 2-D DWT is characterized by three wavelets (see Figure 14):

$$\psi_1(x, y) = \phi(x) \psi(y) \quad (\text{LH wavelet}), \quad (36)$$

$$\psi_2(x, y) = \psi(x) \phi(y) \quad (\text{HL wavelet}), \quad (37)$$

$$\psi_3(x, y) = \psi(x) \psi(y) \quad (\text{HH wavelet}). \quad (38)$$

The LH wavelet is the product of the lowpass function $\phi(\cdot)$ along the first dimension and the highpass (actually a bandpass) function $\psi(\cdot)$ along the second dimension. The HL and HH wavelets are similarly labeled. While the LH and HL wavelets are oriented vertically and horizontally, the HH wavelet has a *checkerboard* appearance — it mixes $+45$ and -45 degree orientations. Consequently, the separable DWT fails to isolate these orientations.

One way to understand why the checkerboard artifact arises in the separable DWT is to look in the frequency

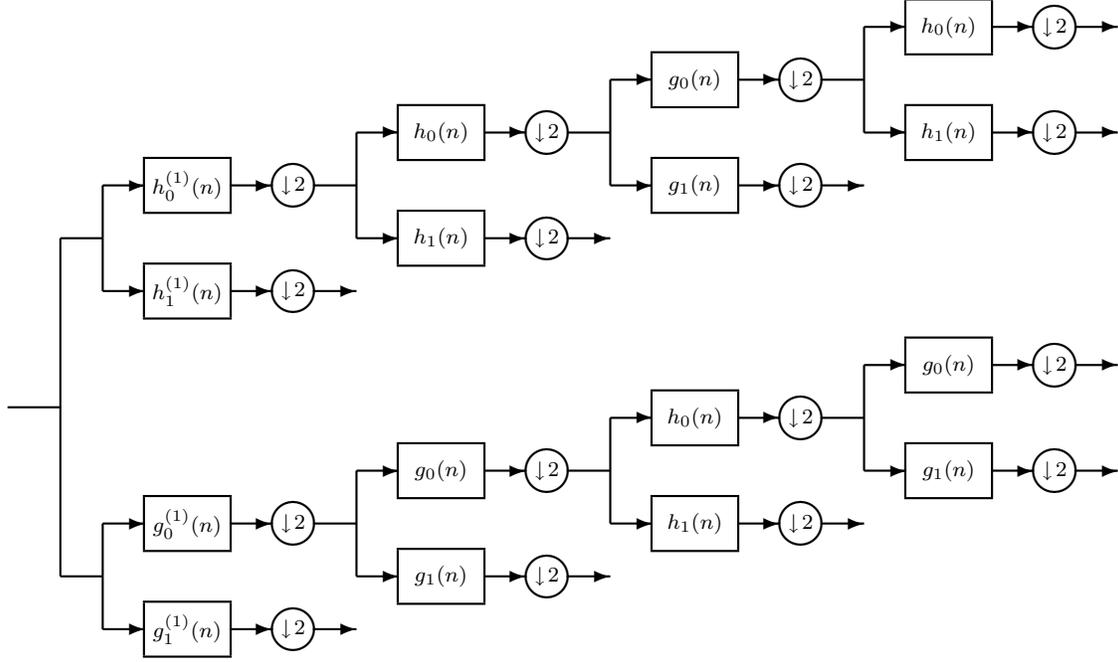
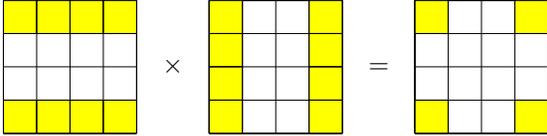


Figure 13: The dual-tree CWT analysis filter bank with alternating filters for each stage (except the first stage). The synthesis filter bank has alternating filters to match the analysis filter bank.

domain. If $\psi(x)$ is a real wavelet and the 2-D separable wavelet is given by $\psi(x, y) = \psi(x) \psi(y)$, then the Fourier spectrum of $\psi(x, y)$ is illustrated by the following idealized diagram:



Since $\psi(x)$ is a real function, its spectrum must be two-sided, and hence it is unavoidable that the 2-D spectrum contains passbands in all four corners of the 2-D frequency plane. Therefore, this wavelet will be unable to distinguish between $+45$ and -45 degree spectral features, and this leads also to the same ambiguity in the space domain.

4.2 2-D dual-tree CWT

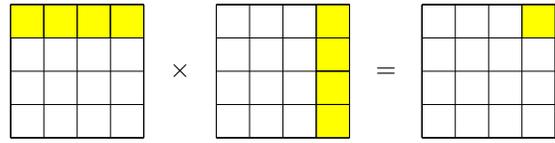
To explain how the dual-tree CWT produces oriented wavelets, consider the 2-D wavelet $\psi(x, y) = \psi(x) \psi(y)$ associated with the row-column implementation of the wavelet transform, where $\psi(x)$ is a complex (approximately analytic) wavelet given by $\psi(x) = \psi_h(x) +$

$j \psi_g(x)$. We obtain for $\psi(x, y)$ the expression

$$\psi(x, y) = [\psi_h(x) + j \psi_g(x)] [\psi_h(y) + j \psi_g(y)] \quad (39)$$

$$= \psi_h(x) \psi_h(y) - \psi_g(x) \psi_g(y) + j [\psi_g(x) \psi_h(y) + \psi_h(x) \psi_g(y)]. \quad (40)$$

The support of the Fourier spectrum of this complex wavelet is illustrated by the following idealized diagram:



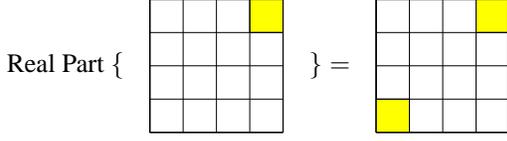
Since the spectrum of the (approximately) analytic 1-D wavelet is supported on only one side of the frequency axis, the spectrum of the complex 2-D wavelet $\psi(x, y)$ is supported in only one quadrant of the 2-D frequency plane. For this reason, the complex 2-D wavelet is oriented.

If we take the real part of this complex wavelet, then we obtain the sum of two separable wavelets

$$\text{Real Part}\{\psi(x, y)\} = \psi_h(x) \psi_h(y) - \psi_g(x) \psi_g(y). \quad (41)$$

Since the spectrum of a real function must be symmetric with respect to the origin, the spectrum of this real wavelet

is supported in two quadrants of the 2-D frequency plane, as illustrated in the following (idealized) diagram:



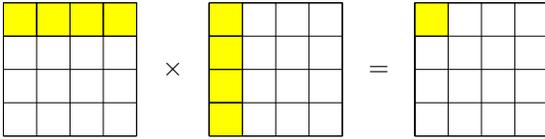
Unlike the real separable wavelet, the support of the spectrum of this real wavelet does not possess the checkerboard artifact and therefore this real wavelet, illustrated in the second panel of Figure 15, is oriented at -45 degrees. Note that this construction depends on the complex wavelet $\psi(x) = \psi_h(x) + j\psi_g(x)$ being (approximately) analytic or, equivalently, on $\psi_g(t)$ being approximately the Hilbert transform of $\psi_h(t)$, $[\psi_g(t) \approx \mathcal{H}\{\psi_h(t)\}]$.

Note that the first term in expression (41), $\psi_h(x)\psi_h(y)$, is the HH wavelet of a separable 2-D real wavelet transform implemented using the filters $\{h_0(n), h_1(n)\}$. The second term, $\psi_g(x)\psi_g(y)$, is also the HH wavelet of a real separable wavelet transform, but one that is implemented using the filters $\{g_0(n), g_1(n)\}$.

To obtain a real 2-D wavelet oriented at $+45$ degrees, consider now the complex 2-D wavelet $\psi_2(x, y) = \psi(x)\psi(y)$ where $\psi(y)$ represents the complex-conjugate of $\psi(x)$ and, as above, $\psi(x)$ is the approximately analytic wavelet $\psi(x) = \psi_h(x) + j\psi_g(x)$. We obtain for $\psi_2(x, y)$ the expression

$$\begin{aligned}\psi_2(x, y) &= [\psi_h(x) + j\psi_g(x)] \overline{[\psi_h(y) + j\psi_g(y)]} \\ &= [\psi_h(x) + j\psi_g(x)] [\psi_h(y) - j\psi_g(y)] \\ &= \psi_h(x)\psi_h(y) + \psi_g(x)\psi_g(y) + \\ &\quad j[\psi_g(x)\psi_h(y) - \psi_h(x)\psi_g(y)].\end{aligned}$$

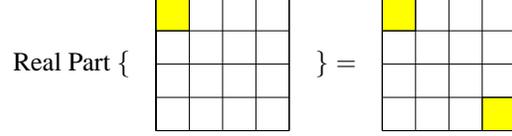
The support in the 2-D frequency plane of the spectrum of this complex wavelet is illustrated by the following idealized diagram:



As above, the spectrum of the complex 2-D wavelet $\psi_2(x, y)$ is supported in only one quadrant of the 2-D frequency plane. If we take the real part of this complex wavelet, then we obtain the real wavelet

$$\text{Real Part}\{\psi_2(x, y)\} = \psi_h(x)\psi_h(y) + \psi_g(x)\psi_g(y), \quad (42)$$

the spectrum of which is supported in two quadrants of the 2-D frequency plane, as illustrated in the following (idealized) diagram:



Again, neither the spectrum of this real wavelet nor the wavelet itself possesses the checkerboard artifact. This real 2-D wavelet is oriented at $+45$ degrees as illustrated in the fifth panel of Figure 15.

To obtain four more oriented real 2-D wavelets we can repeat this procedure on the following complex 2-D wavelets: $\phi(x)\psi(y)$, $\psi(x)\phi(y)$, $\phi(x)\overline{\psi(y)}$, and $\psi(x)\overline{\phi(y)}$; where $\psi(x) = \psi_h(x) + j\psi_g(x)$ and $\phi(x) = \phi_h(x) + j\phi_g(x)$. By taking the real part of each of these four complex wavelets we obtain four real oriented 2-D wavelets, in addition to the two already obtained in (41) and (42). Specifically, we obtain the following six wavelets:

$$\psi_i(x, y) = \frac{1}{\sqrt{2}} (\psi_{1,i}(x, y) - \psi_{2,i}(x, y)), \quad (43)$$

$$\psi_{i+3}(x, y) = \frac{1}{\sqrt{2}} (\psi_{1,i}(x, y) + \psi_{2,i}(x, y)) \quad (44)$$

for $i = 1, 2, 3$, where the two separable 2-D wavelet bases are defined in the usual manner:

$$\psi_{1,1}(x, y) = \phi_h(x)\psi_h(y), \quad \psi_{2,1}(x, y) = \phi_g(x)\psi_g(y), \quad (45)$$

$$\psi_{1,2}(x, y) = \psi_h(x)\phi_h(y), \quad \psi_{2,2}(x, y) = \psi_g(x)\phi_g(y), \quad (46)$$

$$\psi_{1,3}(x, y) = \psi_h(x)\psi_h(y), \quad \psi_{2,3}(x, y) = \psi_g(x)\psi_g(y). \quad (47)$$

We have used the normalization $1/\sqrt{2}$ only so that the sum/difference operation constitutes an orthonormal operation. Figure 15 illustrates the six real oriented wavelets derived from a pair of typical wavelets satisfying $\psi_g(t) \approx \mathcal{H}\{\psi_h(t)\}$. Compared with separable wavelets (see Figure 14), these six wavelets (which are strictly non-separable) succeed in isolating different orientations — each of the six wavelets are aligned along a specific direction and no checkerboard effect appears. Moreover, they cover more distinct orientations than the separable DWT wavelets.

In addition, since the sum/difference operation is orthonormal, the set of wavelets obtained from integer translates and their dyadic dilations form a *frame* (roughly speaking an “overcomplete” basis) [26]. (If the 1-D wavelets $\psi_g(t)$ and $\psi_h(t)$ form orthonormal bases, then the set constitutes a *tight frame*, or a *self-inverting* transform.)

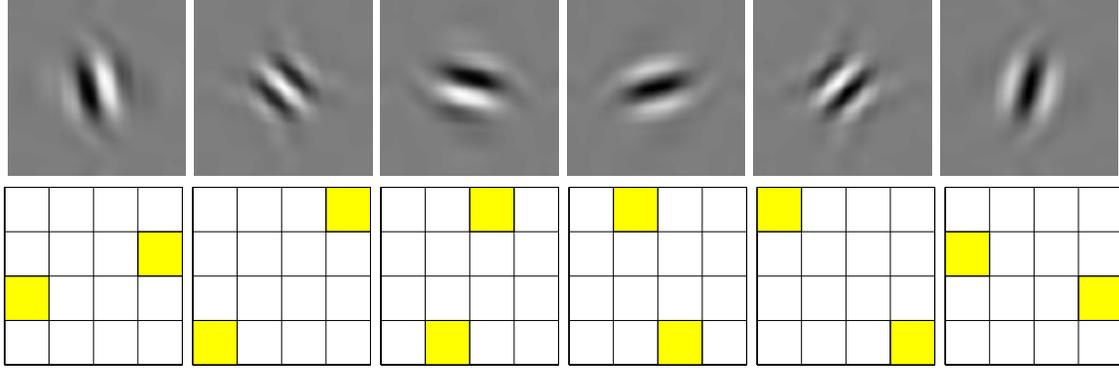


Figure 15: Typical wavelets associated with the real oriented 2-D dual-tree wavelet transform. Top row illustrates the wavelets in the space domain; bottom row illustrates the (idealized) support of the Fourier spectrum of each wavelet in the 2-D frequency plane. The absence of the checkerboard phenomenon is observed in both the spatial and frequency domains.

4.3 Real oriented 2-D dual-tree transform

Since the wavelets in (45)–(47) are all separable, a 2-D wavelet transform based on these six oriented wavelets can be implemented using two real separable 2-D wavelet transforms in parallel. We call this the *real oriented 2-D dual-tree wavelet transform*. The implementation is simple: Use $\{h_0(n), h_1(n)\}$ to implement one separable 2-D wavelet transform; use $\{g_0(n), g_1(n)\}$ to implement another. Applying both separable transforms to the same 2-D data gives a total of six subbands: two HL, two LH, and two HH subbands. To implement the oriented wavelet transform, take the sum and difference of each pair of subbands. The transform is then two-times expansive and free of the checkerboard artifact.

To clarify, suppose that the usual 2-D separable DWT implemented using the filters $\{h_0(n), h_1(n)\}$ is represented by the square matrix \mathbf{F}_{hh} , and suppose that the 2-D separable DWT implemented using the filters $\{g_0(n), g_1(n)\}$ is represented by the square matrix \mathbf{F}_{gg} . (Representing a 2-D transform as a square matrix calls for organizing the 2-D array of pixels into a 1-D vector, but this reorganization is not actually performed in the row-column implementation.) Then the oriented real 2-D dual-tree wavelet transform is represented by the rectangular matrix

$$\mathbf{F}_{2D} = \frac{1}{2} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{hh} \\ \mathbf{F}_{gg} \end{bmatrix}.$$

A (left) inverse of \mathbf{F}_{dt} is then given by

$$\mathbf{F}_{2D}^{-1} = \frac{1}{2} \begin{bmatrix} \mathbf{F}_{hh}^{-1} & \mathbf{F}_{gg}^{-1} \\ -\mathbf{I} & \mathbf{I} \end{bmatrix}.$$

If the two real separable 2-D wavelet transforms are

orthonormal transforms then the transpose of \mathbf{F}_{hh} is its inverse: $\mathbf{F}_{hh}^t \cdot \mathbf{F}_{hh} = \mathbf{I}$, and similarly $\mathbf{F}_{gg}^t \cdot \mathbf{F}_{gg} = \mathbf{I}$. Consequently, the transpose of \mathbf{F}_{2D} is also its inverse: $\mathbf{F}_{2D}^t \cdot \mathbf{F}_{2D} = \mathbf{I}$. That is, the inverse of the oriented 2-D dual-tree wavelet transform can be performed using the transpose of the forward transform. Therefore, the transform satisfies Parseval’s energy theorem and the oriented wavelets form a tight frame [26].

Note that this oriented wavelet transform is non-separable, but it does not have the implementation complexity of a general non-separable transform, nor does it require a solution to a difficult design problem associated with a general non-separable transform. Indeed, the implementation requires only the addition and subtraction of respective subbands of two 2-D separable real wavelet transforms; and it requires no new filter design beyond the 1-D filter design problem of the 1-D dual-tree CWT discussed above.

Like the 1-D dual-tree CWT, the oriented real 2-D dual-tree wavelet transform is still a “dual-tree” wavelet transform and is also two-times expansive. However, it is not in any way a complex transform — the coefficients are not complex, nor should they be interpreted as the real and imaginary parts of complex coefficients. Therefore, while this transform has the benefit of being oriented, it does not share the benefits of an (analytic) complex wavelet transform outlined in Section 1. In particular it will not be approximately shift-invariant.

4.4 Oriented 2-D dual-tree CWT

A 2-D wavelet transform that is both oriented and complex (approximately analytic) can also be easily developed. The *oriented complex 2-D dual-tree wavelet trans-*

form is four-times expansive, but it has the benefit of being both oriented and approximately analytic. It also possesses the full shift-invariant properties of the constituent 1-D transforms. To develop this transform, consider taking the imaginary part of (40) to obtain

$$\text{Imag Part}\{\psi(x, y)\} = \psi_g(x) \psi_h(y) + \psi_h(x) \psi_g(y). \quad (48)$$

The (idealized) support of the spectrum of $\text{Imag Part}\{\psi(x, y)\}$ in the 2-D frequency plane is the same as the spectrum of the real part in (41), and therefore the real 2-D wavelet in (48) is also oriented at -45 degrees. Note that the first term of (48), $\psi_g(x) \psi_h(y)$, is the HH wavelet of a separable real 2-D wavelet transform implemented using the filters $\{g_0(n), g_1(n)\}$ on the *rows*, and the filters $\{h_0(n), h_1(n)\}$ on the *columns* of the image. Similarly, the second term, $\psi_h(x) \psi_g(y)$, is also the HH wavelet of a real separable wavelet transform, but one implemented using the filters $\{h_0(n), h_1(n)\}$ on the *rows* and $\{g_0(n), g_1(n)\}$ on the *columns*. Likewise, we consider also the imaginary parts of $\psi(x) \psi(y)$, $\phi(x) \psi(y)$, $\psi(x) \phi(y)$, $\phi(x) \phi(y)$, and $\psi(x) \phi(y)$; where $\psi(x) = \psi_h(x) + j \psi_g(x)$ and $\phi(x) = \phi_h(x) + j \phi_g(x)$. We then obtain six oriented wavelets given by:

$$\psi_i(x, y) = \frac{1}{\sqrt{2}} (\psi_{3,i}(x, y) + \psi_{4,i}(x, y)), \quad (49)$$

$$\psi_{i+3}(x, y) = \frac{1}{\sqrt{2}} (\psi_{3,i}(x, y) - \psi_{4,i}(x, y)) \quad (50)$$

for $i = 1, 2, 3$, where the two separable 2-D wavelet bases are defined as:

$$\psi_{3,1}(x, y) = \phi_g(x) \psi_h(y), \quad \psi_{4,1}(x, y) = \phi_h(x) \psi_g(y), \quad (51)$$

$$\psi_{3,2}(x, y) = \psi_g(x) \phi_h(y), \quad \psi_{4,2}(x, y) = \psi_h(x) \phi_g(y), \quad (52)$$

$$\psi_{3,3}(x, y) = \psi_g(x) \psi_h(y), \quad \psi_{4,3}(x, y) = \psi_h(x) \psi_g(y). \quad (53)$$

The six real-valued wavelets in (49)–(50) are oriented for the same reason the real-valued wavelets of (43)–(44) are oriented. However, a set of six complex wavelet can be formed by using wavelets (43)–(44) as the real parts, and the wavelets (49)–(50) as the imaginary parts. Figure 16 illustrates a set of six oriented complex wavelets obtained in this way. The real and imaginary parts of each complex wavelet are oriented at the same angle, and the magnitude of each complex wavelet is an approximately circular bell-shaped function.

The matrix representation of the oriented complex 2-D dual-tree wavelet transform clarifies the implementation

of the transform. Let the square matrix \mathbf{F}_{gh} denote the 2-D separable wavelet transform implemented using $g_i(n)$ along the rows and $h_i(n)$ along the columns; and let \mathbf{F}_{hg} denote the usage of $h_i(n)$ along the rows and $g_i(n)$ along the columns. Then the oriented complex 2-D dual-tree wavelet transform is represented by the rectangular matrix

$$\mathbf{F}_{\text{ozd}} = \frac{1}{\sqrt{8}} \begin{bmatrix} \mathbf{I} & -\mathbf{I} & & \\ \mathbf{I} & \mathbf{I} & & \\ & & \mathbf{I} & \mathbf{I} \\ & & \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{hh} \\ \mathbf{F}_{gg} \\ \mathbf{F}_{gh} \\ \mathbf{F}_{hg} \end{bmatrix}.$$

A (left) inverse of \mathbf{F}_{ozd} is then given by

$$\mathbf{F}_{\text{ozd}}^{-1} = \frac{1}{\sqrt{8}} \begin{bmatrix} \mathbf{F}_{hh}^{-1} & \mathbf{F}_{gg}^{-1} & \mathbf{F}_{gh}^{-1} & \mathbf{F}_{hg}^{-1} \\ \mathbf{I} & \mathbf{I} & & \\ -\mathbf{I} & \mathbf{I} & & \\ & & \mathbf{I} & \mathbf{I} \\ & & \mathbf{I} & -\mathbf{I} \end{bmatrix}. \quad (54)$$

If the individual wavelet transforms are orthonormal transforms then the inverse in (54) is exactly the transpose of the forward transform, and it therefore represents a tight frame.

If the vector \mathbf{x} represents a real-valued image, then

$$\mathbf{w}_1 = \frac{1}{2} \begin{bmatrix} \mathbf{I} & -\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{hh} \\ \mathbf{F}_{gg} \end{bmatrix} \mathbf{x}$$

represents the real part of the oriented complex transform and

$$\mathbf{w}_2 = \frac{1}{2} \begin{bmatrix} \mathbf{I} & \mathbf{I} \\ \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{gh} \\ \mathbf{F}_{hg} \end{bmatrix} \mathbf{x}$$

represents the imaginary part. In this implementation the real and imaginary parts are stored separately. The complex wavelet coefficients are $\mathbf{w}_1 + j \mathbf{w}_2$.

If the transform is applied to a complex-valued image then the complex coefficients should be formed explicitly as follows:

$$\mathbf{F}_{\text{czd}} = \frac{1}{4} \begin{bmatrix} \mathbf{I} & & j\mathbf{I} & & & \\ & \mathbf{I} & & j\mathbf{I} & & \\ \mathbf{I} & & -j\mathbf{I} & & & \\ & \mathbf{I} & & -j\mathbf{I} & & \end{bmatrix} \begin{bmatrix} \mathbf{I} & -\mathbf{I} & & \\ \mathbf{I} & \mathbf{I} & & \\ & & \mathbf{I} & \mathbf{I} \\ & & \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{F}_{hh} \\ \mathbf{F}_{gg} \\ \mathbf{F}_{gh} \\ \mathbf{F}_{hg} \end{bmatrix}$$

and

$$\mathbf{F}_{\text{czd}}^{-1} = \frac{1}{4} \begin{bmatrix} \mathbf{F}_{hh}^{-1} & \mathbf{F}_{gg}^{-1} & \mathbf{F}_{gh}^{-1} & \mathbf{F}_{hg}^{-1} \\ \mathbf{I} & \mathbf{I} & & \\ -\mathbf{I} & \mathbf{I} & & \\ & & \mathbf{I} & \mathbf{I} \\ & & \mathbf{I} & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{I} & & \mathbf{I} & \\ & \mathbf{I} & & \mathbf{I} \\ -j\mathbf{I} & & j\mathbf{I} & \\ & -j\mathbf{I} & & j\mathbf{I} \end{bmatrix}.$$

Note that the oriented 2-D dual-tree CWT (applied to real or complex data) requires four separable wavelet

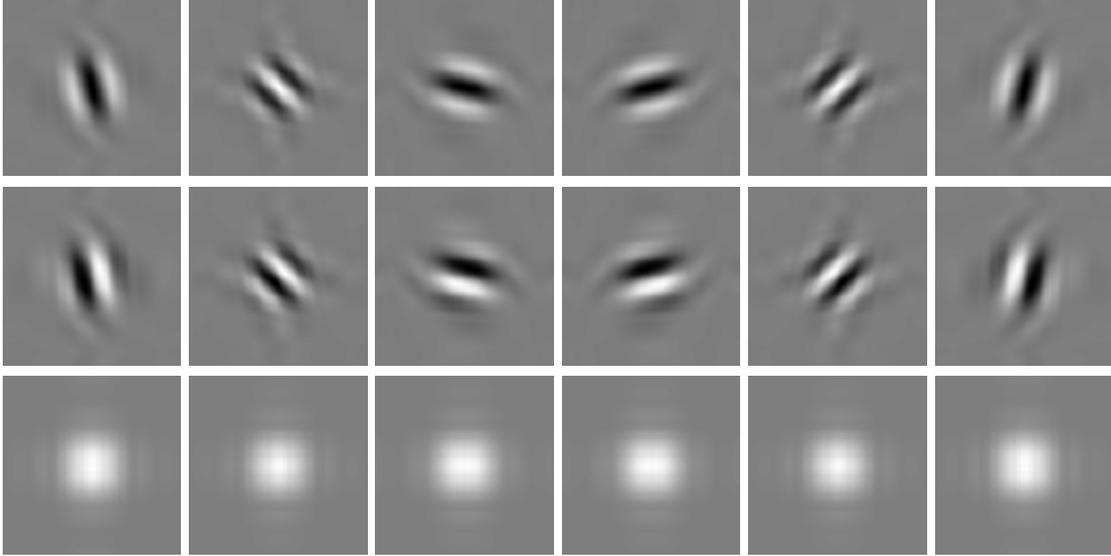


Figure 16: Typical wavelets associated with the oriented 2-D dual-tree CWT. Top row illustrates the real part of each complex wavelet; second row illustrates the imaginary part; and third row illustrates the magnitude.

transforms in parallel, and so it is no longer strictly a “dual-tree” wavelet transform. However, we still refer to it as such for convenience and because it is derived from the 1-D dual-tree CWT. Similarly, while the wavelets are oriented, approximately analytic, and nonseparable, the implementation is still very efficient, requiring only the addition and subtraction of respective subbands of four 2-D separable wavelet transforms.

4.5 Links with the 2-D Gabor transform

Gabor analysis is frequently used in image processing and pattern analysis. A 2-D *Gabor function* is a 2-D Gaussian window multiplied by a complex sinusoid

$$f(x, y) = e^{-((x/\sigma_1)^2 + (y/\sigma_2)^2)} e^{-j(\omega_x x + \omega_y y)}.$$

Gabor functions are optimally concentrated in the space-frequency plane. Certain image analysis algorithms use Gabor functions as the impulse response of a set of 2-D filters [40]. By varying the parameters ω_x and ω_y , the orientation of the Gabor function can be adjusted; by varying σ_1 and σ_2 the spatial extent and aspect ratio of the function can be adjusted. Some Gabor-based image processing algorithms are designed to use both magnitude and phase information of Gabor-filtered images.

The 2-D dual-tree wavelets illustrated in Figure 16 resemble 2-D Gabor functions to some degree. However, in contrast to analysis by Gabor functions, the 2-D dual-tree

CWT is based on FIR filter banks with a fast invertible implementation. A typical Gabor image analysis is either expensive to compute, is non-invertible, or both. With the 2-D dual-tree CWT, many ideas and techniques from Gabor analysis can be leveraged into wavelet-based image processing.

The oriented complex wavelets illustrated in Figure 16 also resemble to some degree the set of 2-D functions computed by Olshausen and Field [75]. They proposed that parts of biological visual systems are based on the efficient representation of natural images by an overcomplete set of 2-D functions. They proposed an optimality criterion based on sparsity, developed an iterative numerical algorithm, and obtained as a solution a remarkable set of 2-D functions exhibiting interesting properties: the functions are mostly well oriented and occur at various scales. Their result confirms to some degree the notion that oriented wavelet and wavelet-like transforms are natural for image processing applications.

4.6 Extensions to higher dimensions

The dual-tree CWT can be extended to higher dimensions than two using the procedure described above. In the d -dimensional case, the oriented dual-tree *real* wavelet transform is expansive by 2^{d-1} ; the oriented *complex* wavelet transform is expansive by 2^d . Importantly, the checkerboard artifact of the conventional separable DWT becomes ever more serious in higher dimensions. Cor-

respondingly, the gain provided by using the oriented wavelet transform grows with the dimension d . The 3-D dual-tree wavelet transforms shows promise for processing medical volume data and video sequences [90]. Application of complex and oriented 3-D wavelet transforms to seismic analysis is described in [109]. A higher-D generalization of the CWT to a *hyper-complex* wavelet transform (based on quaternions and octonions) has been introduced in [17–19].

5 Using the dual-tree CWT

The key advantages of the dual-tree CWT over the DWT are its shift invariance and directional selectivity. This means that the squared magnitude of a given complex wavelet coefficient provides an accurate measure of spectral energy at a particular location in space, scale, and orientation. It also means that CWT-based algorithms will automatically be almost shift-invariant, thus reducing many of the artifacts of the critically-sampled DWT. Here we illustrate some additional attractive properties of the CWT along with some prototypical applications.

5.1 Near shift invariance

One way to illustrate the near shift-invariance of the dual-tree CWT is to observe how the projection of a signal onto a certain scale varies as the signal translates. The projection of a signal onto scale j can be computed by reconstructing the signal from only the wavelet coefficients in subband j . Figure 17 (top-left panel) shows a simple pulse signal $x(n)$ and its reconstruction from wavelet coefficients at the 3rd scale level of the critically sampled DWT and the dual-tree CWT. The top-right panel of the figure shows the same signal translated by 3 samples and the corresponding reconstructions from level 3. Comparing the left and right panels of Figure 17 we see that the DWT-reconstructed signal varies significantly with translations of the signal. However, the CWT-reconstructed signal maintains its shape, illustrating the near shift-invariance of the dual-tree CWT. This property of the CWT greatly simplifies wavelet-based modeling, processing, and other applications.

The source of the near shift invariance property can be understood in two different ways. First, since the real and imaginary wavelets are Hilbert transforms of each other (90° out of phase), the real and imaginary wavelet coefficients interpolate each other. Second, since we use two trees, the effect of the decimation by two at each scale is diminished, which greatly reduces the amount of aliasing.

The near shift-invariance of the dual-tree CWT can be quantified. The measure of shift dependence defined in Equation (5) of [57] is based on the ratio of the energy of the aliased components of the transfer function through a given subband to the energy of the unaliased components. A truly shift invariant transform has the property that the signal path through any single subband of the transform and its inverse may be characterized by a unique z transfer function, which is unaffected by the down and up sampling within the transform.

5.2 Local Hilbert transform

The envelope of a real signal can be computed using the Hilbert transform to create a complex-valued analytic signal; the magnitude is the sought envelope. However, a time- or frequency-based Hilbert transform may produce undesired behavior around transients of the signal due to the slow decay of the impulse response of the ideal Hilbert transformation (61). A *local Hilbert transform* can be computed in the complex wavelet domain simply by multiplying the CWT coefficients by j . As a bonus, the CWT-based local Hilbert transform can be efficiently implemented by a continuously running filter bank. An example is shown in Figure 18. Multidimensional CWT-based local Hilbert transforms have been proposed in [109] for seismic data analysis. An interesting feature of CWT-based Hilbert transforms is that the transition region around zero frequency may be made arbitrarily sharp by adding additional levels of wavelet decomposition. This requires a negligible increase in computation cost, but it does add extra delay.

5.3 Near rotation invariance

The directionality of the 2-D CWT renders it nearly rotation invariant in addition to nearly shift invariant. Figure 19 illustrates the image obtained by reconstruction from only one level of the real DWT and dual-tree CWT for a test image with a sharp edge on a hyperbolic trajectory. The ringing and aliasing artifacts in the DWT coefficients that change with the edge orientation are not present in the CWT coefficients.

5.4 Image rotation

While there are more direct methods for image rotation (via image interpolation in the pixel domain) it is interesting to note that it is possible to do this in the wavelet domain using the dual-tree CWT. This relies on the uniqueness of the z transfer functions with shift, mentioned in

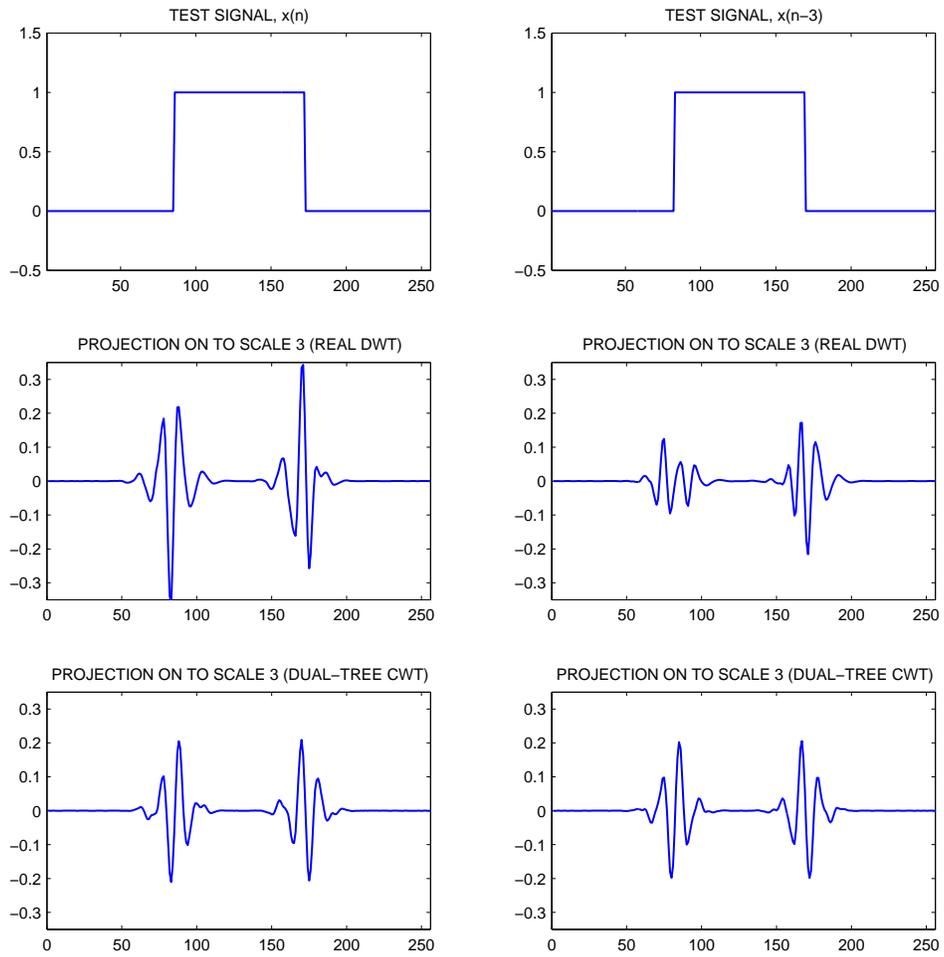


Figure 17: A signal $x(n)$ and its shifted version $x(n - 3)$ (top panels) and its reconstruction from wavelet coefficients at scale level 3 of the real DWT (middle panels) and dual-tree CWT (lower panels). The CWT is more nearly shift-invariant than the DWT.

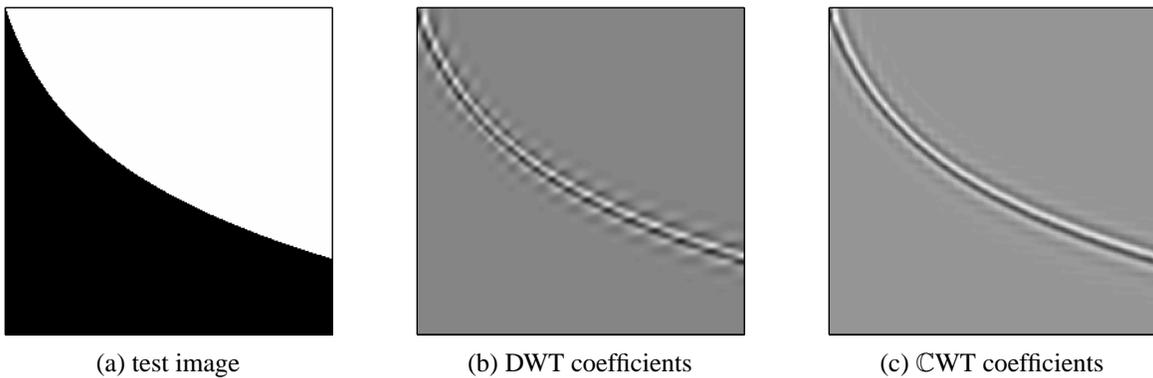


Figure 19: Near rotation invariance of the CWT. (a) Test image with sharp edge on hyperbolic trajectory. (b) When the test image is reconstructed from one level of the DWT coefficients, ringing and aliasing effects are apparent. (c) The reconstruction of the image from one level of the CWT does not exhibit these phenomena.

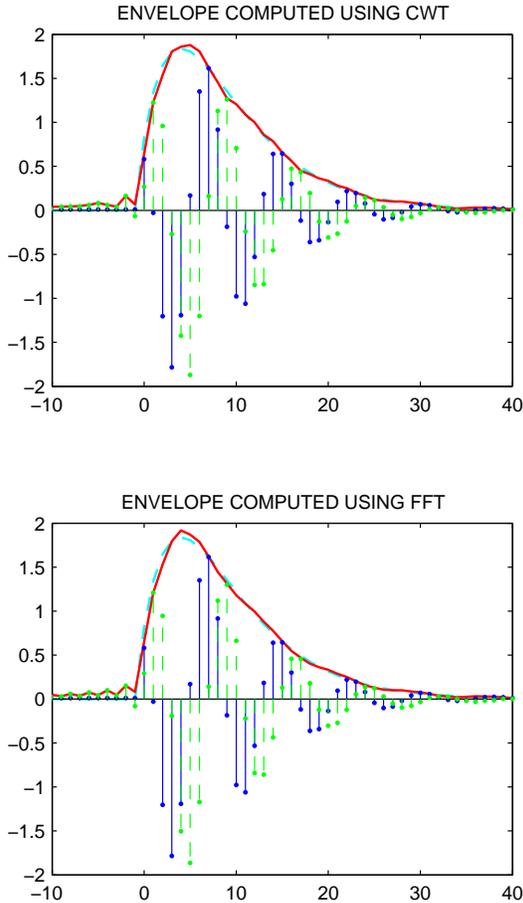


Figure 18: The dual-tree CWT provides a way to perform a local approximate Hilbert transform. The FFT gives similar results, but it requires overlapped block implementation for real-time data, whereas the dual-tree CWT can be implemented as a continuously running filter bank. In each case, the input waveform, $x(t) = t \exp(-0.2t) \cos(0.8t)$ for $t = 0, 1 \dots 40$, is shown as a blue stem plot, and its local Hilbert transform, $y(t)$, as a green dashed stem plot. The ‘true’ envelope, $t \exp(-0.2t)$, is shown as a cyan dashed line and the envelope extracted by $|x(t) + jy(t)|$ is shown as a red solid line.

section 5.1, and the resulting interpolability of each sub-band. By shifting the complex coefficients in each sub-band independently, we can rotate an image by small angles. This is achieved by a band-limited interpolation process, in which the complex coefficients (a) are first de-rotated by the band center-frequency, (b) are then interpolated using the MATLAB command `interp2`, and (c) are then re-rotated back up to their original frequency range. For example, Figure 20 illustrates the Barbara image and a 5.7° (0.1 radians) rotated version. Note the blurring effects in the corners where there would be undefined pixels in a space-domain rotation scheme. This technique may also be used to achieve other arbitrary smoothly varying displacements, provided that any rotation components are small enough that there is little energy transfer between directional subbands (i.e. less than about 10 degrees).

5.5 Estimating image geometrical structure

The shift and rotation invariance properties of the CWT can also be harnessed to compute accurate and efficient estimates of the geometrical structure in images, namely the strength, orientation, and offset of image edges, ridges, and other singularities.

Consider the edge segment depicted in Figure 21(a) and fix the scale of the CWT so that the wavelets have roughly this support size. Then, as the orientation θ and offset r of the edge change, so do the magnitude and phase of the CWT coefficients [57, 81, 113]. In particular, as we see from Figure 21(b) the magnitudes of the CWT coefficients peak as the edge orientation θ approaches their orientation; we can estimate the edge orientation to within approximately 2° error by simply interpolating between these response curves [81]. Moreover, the edge offset r can be estimated directly from the phase of the CWT coefficient with largest magnitude. Finally, this same largest coefficient indicates the strength of the edge. Figure 22 illustrates this procedure on a test image.

The related problem of predicting the phase of a complex coefficient from one scale to the next has been addressed for 1-D signals in [82, 117].

5.6 Estimating local displacement

Local displacement (motion) between two images can be estimated from the change of phase of CWT coefficients from one image to the next. As in the single image case in Section 5.5, at each position and orientation, the change $\Delta\phi_a$ of the phase of a complex wavelet coefficient is approximately linearly proportional to the displacement in a direction orthogonal to the subband orientation. From the

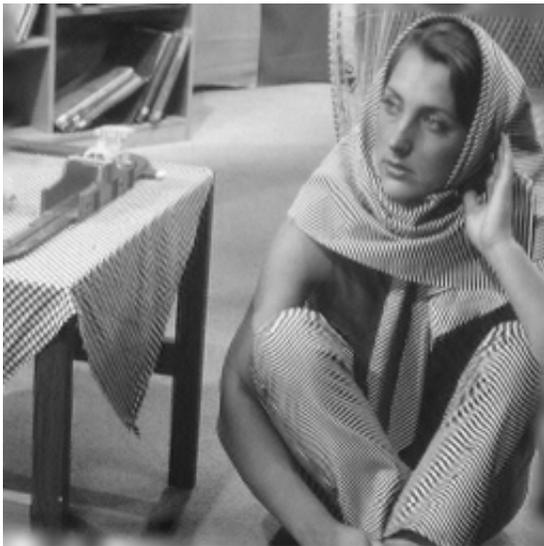
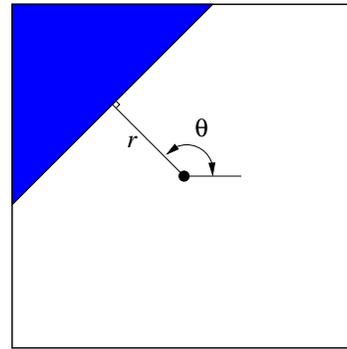
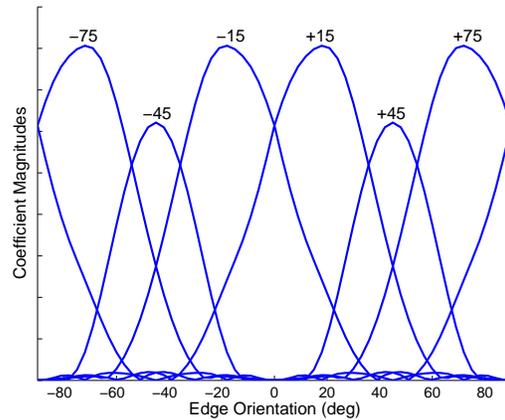


Figure 20: \mathbb{C} WT-based image rotation by 5.7° by independently shifting the complex wavelet coefficients in each sub-band.



(a) edge segment



(b) \mathbb{C} WT magnitude responses

Figure 21: (a) Image segment with an edge singularity at orientation θ and offset from center r . (b) Magnitude responses of the \mathbb{C} WT coefficients of this segment as a function of θ .

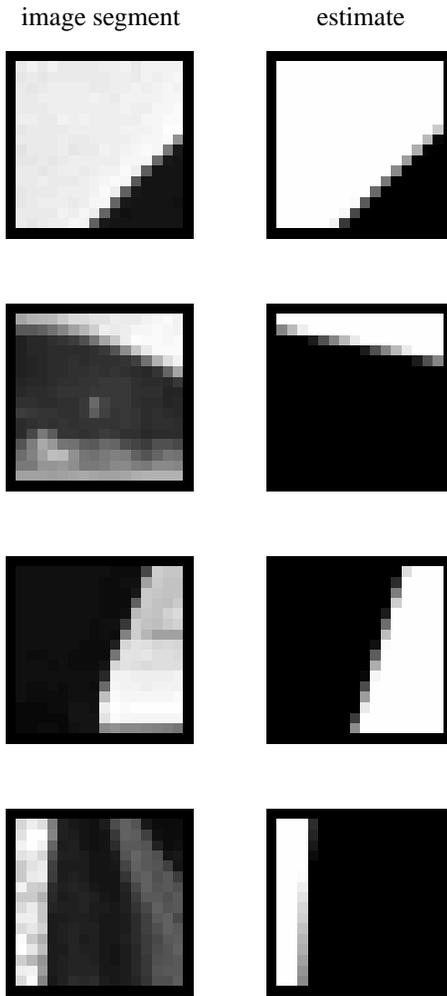


Figure 22: At top the Cameraman test image and four segments with strong edges. Below, zooms of the segments plus idealized edges formed with the parameters estimated from the CWT magnitude and phase. (No attempt is made to match the texture within the segment, only the edge parameters.)

six $\Delta\phi_d$ values (one for each subband), a best-fit displacement vector and associated confidence ellipse can be estimated. Propagation of vectors from coarse to fine scales can then provide resilience to aperture problems. Further details are given in [19, 67, 81, 113]. It is also appropriate to use more complicated strategies for phase-based displacement estimation with the CWT such as in [47].

5.7 Denoising

Basic wavelet-based image denoising algorithms use the DWT and hard or soft thresholding. Substantial performance improvements can be obtained through other transforms (such as the undecimated DWT [23, 63], steerable pyramid [95], or curvelet transform [100]) and through more effective, possibly adaptive, non-linearities based on statistical models for the wavelet coefficients [24, 72, 78].

The CWT can give a substantial performance boost to DWT-noise reduction algorithms. When thresholding the complex-valued coefficients of the CWT it is typically more effective to apply the nonlinearity to the *magnitude* rather than to the real and imaginary parts separately. Since the coefficient magnitudes are slowly varying and free of aliasing distortion, this results in a nearly shift-invariant denoising algorithm. Also, denoising algorithms based on statistical models of wavelet coefficients can be more effective for the CWT than for the real DWT because the magnitudes of the coefficients are more strongly dependent in inter-scale and intra-scale neighborhoods [82, 83].

In this example, the 512×512 8-bit gray-scale *Barbara* image was corrupted by additive Gaussian noise with $\sigma_n = 15$. Denoising with the data-driven locally adaptive bishrink algorithm of [91] was performed using both the critically-sampled separable DWT and the dual-tree CWT. The PSNRs for this noise level are 29.85 dB and 31.27 dB respectively. Cropped portions of the images are illustrated in Figure 23. The improved performance from using directionally selective and shift-invariant filters is clear. The effective performance of several other denoising algorithms using the CWT have also been described [22, 83, 118].

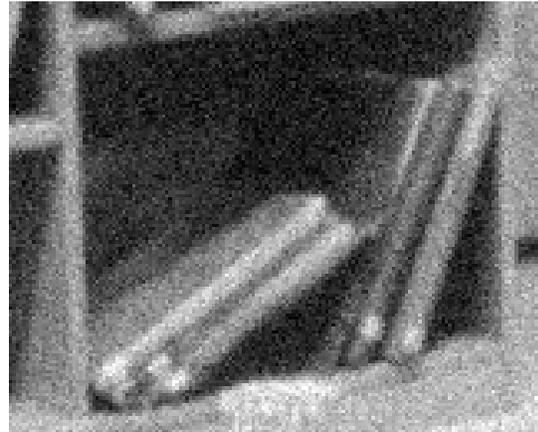
Volume and video denoising can be performed with a 3-D version of the dual-tree CWT [12, 90, 93].

5.8 Additional applications

The dual-tree CWT is suitable for numerous other applications as well, including image segmentation [83, 92], classification [80], deconvolution [29, 51], image sharpening [94], motion estimation [67], coding [79, 97, 115], watermarking [35, 66], texture analysis and synthesis



(a) Noise-free image.



(b) Noisy Image.



(a) Denoised using separable real DWT.



(b) Denoised using dual-tree CWT.

Figure 23: Denoising example using the locally adaptive bishrink algorithm with the critically-sampled real DWT and the dual-tree CWT. A cropped section of the images are shown.

[28, 46, 48], feature extraction [60, 65], seismic imaging [73], and the extraction of evoked potential responses in EEG signals [16]. Complex wavelet transforms (not specifically the dual-tree CWT) have been used recently for measuring image similarity [116].

6 Related Work

There has been substantial work on transforms that are some combination of: multiscale, directional, complex, analytic, nearly shift invariant, overcomplete, and so on. The following gives a brief a brief and non-exhaustive overview of some of them.

(Approximately) analytic continuous wavelet transforms. In their seminal work on the continuous wavelet transform, Grossman and Morlet emphasized complex analytic (exact and approximate) wavelets [45]. Indeed, the Morlet wavelet is complex-valued and approximately analytic. This work in continuous wavelet transforms was continued by Antoine [6, 7] and used for the development of directional wavelets by Vandergheynst et al. [110]. Analytic wavelet transforms and discrete implementations were also used by Abry and Flandrin [3–5] for turbulence analysis, where the quadrature properties of the wavelets were exploited.

Complex filter banks. Complex forms of the discrete wavelet transform were mentioned by Daubechies [27], and complex Daubechies wavelets were studied in depth by Lina [11, 64]. Other complex-valued filter banks have been developed by Gao, Nguyen, and Strang [42, 123]. However, while these solutions are complex-valued, they are not approximately analytic, as noted in Section 2.3.

Directional transforms. Bamberger, Smith, Hong, and Rosiles have developed critically sampled directional 2-D filter banks [10, 49, 84]. Do and Vetterli have developed the *contourlet* transform which can be critically-sampled or slightly over-complete [30–32]. The *curvelet* transform, developed by Candes and Donoho, is an overcomplete directional multiscale transform that is very effective for representing edges in images [15, 100]

Generalizations of the dual-tree CWT. Chaux et al. have developed the M -band dual-tree CWT, generalizing the delay condition for the Hilbert pair property in [21]. Gopinath introduced the *phaselet transform* [43], where more than two critically-sampled DWTs are used together. In this transform, each of M lowpass filters are offset from each other by increments of $1/M$ samples, a generalization of the half-sample delay condition. Another generalization is the double-density dual-tree CWT [89] where two over-sampled (double-density [86]) DWTs are used together. This is further generalized in [44] and [2]. An-

other type of generalization in higher dimensions is the hyper-complex wavelet transform [17–19]. A recently developed complex wavelet transform is also the *RI-spline* [52].

Approximately analytic complex directional transforms. The closest alternative to the dual-tree CWT is probably the complex (approximately) analytic form of the *steerable pyramid* [95, 96]. Simoncelli has used this transform for image denoising and texture analysis and synthesis. Malvar has described complex lapped transforms [70, 71]. Similar transforms have been used for motion estimation [119, 120].

Other recent research activity in the development of complex directional multiscale transforms has focused on the development of critically sampled (non-redundant) implementations, for example by Ates and Orchard, Hua, Spaendonck, and Fernandez et. al. [8, 9, 39, 50, 108, 109]. In a critically-sampled transform, it is difficult to achieve the near shift-invariance of the dual-tree CWT. However, such transforms are promising for image compression.

7 Conclusions

The dual-tree complex wavelet transform (CWT) is a valuable enhancement of the traditional real wavelet transform that is nearly shift invariant and, in higher dimensions, directionally selective. Since the real and imaginary parts of the dual-tree CWT are, in fact, conventional real wavelet transforms, the CWT can benefit from the vast theoretical, practical, and computational resources that have been developed for the standard DWT. For example, software and hardware developed for implementation of the real DWT can be used directly for the CWT. But, in addition, the magnitude and phase of CWT coefficients can be exploited to develop new effective wavelet-based algorithms, especially for applications for which the DWT is unsuited or underperforms.

MATLAB software for the dual-tree complex wavelet transform (and related algorithms) is available at the following locations on the web: <http://taco.poly.edu/WaveletSoftware/>, <http://www-sigproc.eng.cam.ac.uk/~ngk/>, and <http://dsp.rice.edu/>.

8 Acknowledgments

Thanks to Justin Romberg and Michael Wakin for providing Figures 19, 21, and 22. IS thanks ONR for support of this work under grant N00014–03–1–0217. RGB thanks NSF grant FMM 04–520, ONR grant N00014–02–1–0353, AFOSR grant FA9550–04–1–0148, and the Texas Instru-

ments Leadership University Program. NGK thanks Trinity College Cambridge, the UK EPSRC, the UK DTC on Data and Information Fusion, and EU projects MOUMIR and MUSCLE.

A Sidebar: Real-valued Discrete Wavelet Transform and Filter Banks

The discrete wavelet transform (DWT) of (1)-(3) is intimately intertwined with the iterated two-band filter bank (FB) tree structures of Figure 24 [68]. The forward DWT, implemented with the analysis FB of Figure 24(a), computes the scaling and wavelet coefficients $c(n)$ and $d(j, n)$. The input signal is the uniformly spaced samples of a continuous-time signal $x_a(t)$ [$x(n) = x_a(nT)$] or a prefiltered version of them [104]. In many (perhaps most) applications, $x(n)$ is the discrete data itself. For the inverse DWT, the scaling and wavelet coefficients are input to the synthesis FB of Figure 24(b) to produce the signal $y(n)$. The wavelet coefficients $d(j, n)$ in Figure 24 are labeled so that the coarsest scale is denoted by $j = 0$ and j increases for finer scales. In the continuous-time limiting case, the scale index j increase to infinity.

Here we denote the analysis filters by $h_0(n)$ and $h_1(n)$, and the synthesis filters by $\tilde{h}_0(n)$ and $\tilde{h}_1(n)$. For the analysis and synthesis filter banks to represent a forward and inverse wavelet transform, it is necessary that the *perfect reconstruction* (PR) condition be satisfied: $y(n) = x(n)$, or more generally $y(n) = x(n - n_o)$.

Assuming the analysis and synthesis filters are real FIR filters, the perfect reconstruction condition can be satisfied if $h_0(n) * \tilde{h}_0(n)$ is a lowpass *halfband* filter [74, 99, 111]. Specifically, if we define the product filter

$$p(n) := h_0(n) * \tilde{h}_0(n)$$

then for perfect reconstruction (with a delay of n_o samples) it is necessary that

$$p(2n + n_o) = \delta(n) = \begin{cases} 1 & n = 0 \\ 0 & n \neq 0 \end{cases} \quad (55)$$

where the two highpass filters are given by

$$h_1(n) = (-1)^{n+d} \tilde{h}_0(n - d), \quad (56)$$

$$\tilde{h}_1(n) = -(-1)^{n+d} h_0(n + d) \quad (57)$$

and d is an even (or odd) integer when n_o is an odd (or even) integer. When n_o is odd, d can be zero, which simplifies the expressions for the highpass filters.

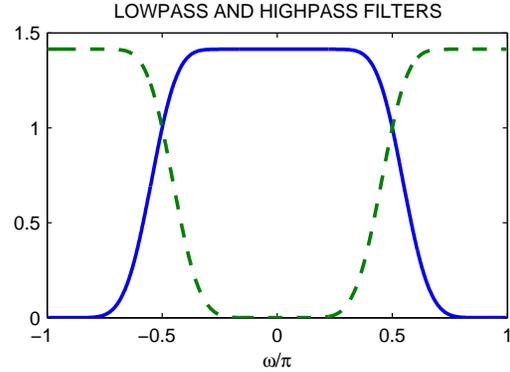


Figure 25: Magnitude frequency responses $|H_0(e^{j\omega})|$ (solid) and $|H_1(e^{j\omega})|$ (dashed) of the real Daubechies lowpass and highpass filters of length 10.

Taking the discrete-time Fourier transform (DTFT), an equivalent condition in terms of the filter frequency responses is

$$|H_0(e^{j\omega})|^2 + |H_1(e^{j\omega})|^2 = 2. \quad (58)$$

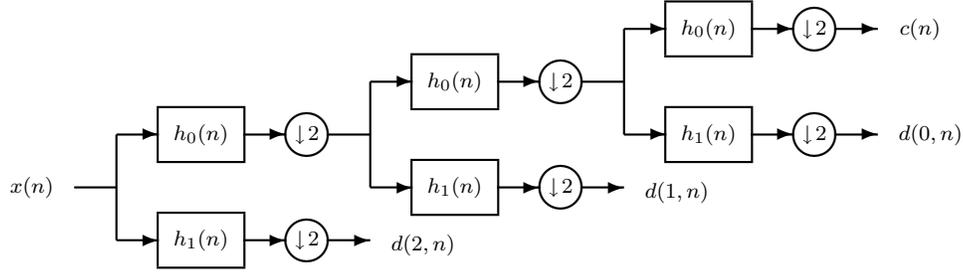
Figure 25 illustrates $|H_0(e^{j\omega})|$ and $|H_1(e^{j\omega})|$ of the lowpass and highpass Daubechies filters of length 10 [27].

Since the analysis FB does not expand the total data rate, we say that it is *critically sampled*. Consequently, for finite length input data, the analysis FB can be viewed as a linear transformation with a square real matrix \mathbf{F} taking the vector \mathbf{x} of signal samples to the vector \mathbf{w} of scaling and wavelet coefficients via $\mathbf{w} = \mathbf{F}\mathbf{x}$. When the transform is perfect reconstruction, we have $\mathbf{x} = \mathbf{F}^{-1}\mathbf{w}$

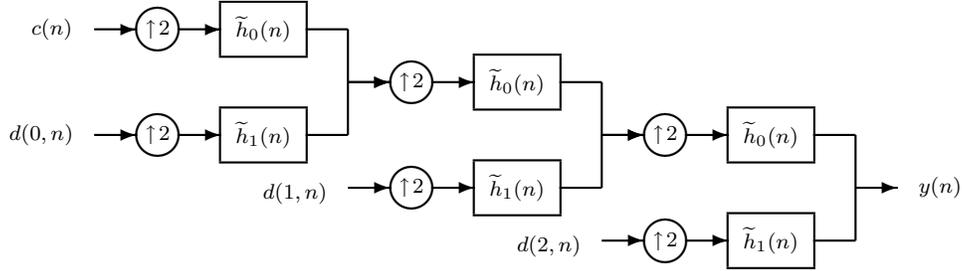
For an orthonormal wavelet transform,⁶ the transform matrix \mathbf{F} satisfies $\mathbf{F} \cdot \mathbf{F}^t = \mathbf{F}^t \cdot \mathbf{F} = \mathbf{I}$; that is, the transpose of \mathbf{F} is also its inverse. It can be shown that the analysis and synthesis filter bank represent an orthonormal transform if the synthesis filters are the time-reversed versions of the analysis filters: $\tilde{h}_0(n) = h_0(L - n)$ and $\tilde{h}_1(n) = h_1(L - n)$ for some L . In this case, the product filter $p(n)$ is the autocorrelation of $h_0(n)$.

Additional constraints on the filters can force orthogonality to low-order polynomials — *vanishing moment* conditions [27] — which is useful for representing smooth and piecewise smooth signals, and *finite time support*, that is, that the wavelet equals zero outside of some time interval. Finite support is extremely useful for wavelet-based signal processing, since it limits the extent to which a signal feature can affect the wavelet coefficients.

⁶When \mathbf{F} is complex, then it represents a unitary transform with $\mathbf{F} \cdot \mathbf{F}^* = \mathbf{F}^* \cdot \mathbf{F} = \mathbf{I}$ where \mathbf{F}^* is the conjugate (hermitian) transpose of \mathbf{F} .



(a) DWT analysis filter bank



(b) DWT synthesis filter bank

Figure 24: Filter bank trees implementing the (a) forward (analysis) and (b) inverse (synthesis) discrete wavelet transform (DWT).

The (analysis) wavelet $\psi(t)$ associated with these filters is given by

$$\psi(t) = \sqrt{2} \sum_n h_1(n) \phi(2t - n) \quad (59)$$

where $\phi(t)$ is called the scaling function and is given implicitly by

$$\phi(t) = \sqrt{2} \sum_n h_0(n) \phi(2t - n). \quad (60)$$

The synthesis wavelet and scaling functions, $\tilde{\psi}(t)$ and $\tilde{\phi}(t)$, are given by the same equations, but using $\tilde{h}_i(n)$ instead of $h_i(n)$. In the orthonormal case, the synthesis wavelet is the time-reversed version of the analysis wavelet. Equation (60), called the *dilation equation*, is a central equation in the theory of wavelet bases and has been studied extensively since the advent of wavelet transforms [103]. We note here that a well defined solution to the dilation equation exists only when $h_0(n)$ is a lowpass filter with $H_0(z = -1) = 0$. From equations (59) and (60), the wavelets are fully determined by the filters $h_0(n)$ and $h_1(n)$, so therefore, the design of a wavelet $\psi(t)$ satisfying specific properties is equivalent to the design of filters $h_i(n)$ satisfying specific properties. For example, if the filters have finite support, then so do the wavelet

and scaling function. And, if the filters have vanishing moments, then so do the wavelet and scaling function.

B Sidebar: The Hilbert Transform and Analytic Signal

A fundamental problem appearing in many signal processing and communications applications is that of extracting the *amplitude* $a(t)$ and instantaneous *phase* $\rho(t)$ of a real, modulated signal

$$x(t) = a(t) \cos(\rho(t)).$$

Retrieval of $a(t)$ is ill-posed when $\cos(\rho(t)) \approx 0$. A clever solution sidesteps this problem by making the real signal $x(t)$ complex through the *Hilbert transform* [77]

$$(\mathcal{H}x)(t) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{x(\tau)}{t - \tau} d\tau. \quad (61)$$

Note that the impulse response of the Hilbert transform is

$$h_{\mathcal{H}}(t) = \frac{1}{\pi t}$$

which decays slowly. If the underlying amplitude function $a(t)$ is assumed to be relatively narrowband com-

pared with $x(t)$, then the *analytic signal*,

$$x_a(t) = x(t) + j(\mathcal{H}x)(t),$$

where $j = \sqrt{-1}$, becomes

$$x_a(t) = a(t) \cos(\rho(t)) + j a(t) \sin(\rho(t)) = a(t) e^{j\rho(t)}.$$

Estimation of the magnitude $a(t)$ is now well-posed and straightforward via

$$|a(t)| = |x_a(t)|.$$

The Hilbert transform has several useful and interesting properties. First, $x(t)$ and $(\mathcal{H}x)(t)$ have the same magnitude function $a(t)$ but phases that are shifted by 90° . Second, the frequency response of the filter corresponding to (61) is

$$H_{\mathcal{H}}(\Omega) = \begin{cases} -j, & \Omega > 0 \\ 0, & \Omega = 0 \\ j, & \Omega < 0. \end{cases} \quad (62)$$

Thus, the overall filter corresponding to the transformation $x(t) \rightarrow x_a(t)$ suppresses negative frequencies

$$H_a(\Omega) = 1 + jH_{\mathcal{H}}(\Omega) = \begin{cases} 2, & \Omega > 0 \\ 1, & \Omega = 0 \\ 0, & \Omega < 0. \end{cases} \quad (63)$$

Since $x(t)$ is real, its Fourier transform $X(\Omega)$ has conjugate symmetry; the filter $H_a(\Omega)$ produces $X_a(\Omega) = 2X(\Omega)$ for $\Omega > 0$ and sets $X_a(\Omega) = 0$ for $\Omega < 0$. Note that due to the discontinuity of $H_{\mathcal{H}}(\Omega)$ at $\Omega = 0$ a transition band must be allowed in practice.

Third, when the phase function is linear such that $\rho(t) = \Omega_0 t$, a time shift of the real signal manifests itself as a time shift of the amplitude and a phase shift of the phase. That is, if $y(t) = x(t - t_0)$, then

$$y_a(t) = x_a(t - t_0) = a(t - t_0) e^{j\rho(t)} e^{-j\Omega_0 t_0}.$$

The definitions of Hilbert transform and analytic signal are similar for discrete-time signals.

References

- [1] A. Abbas and T. Tran. Fast approximations of the orthogonal dual-tree wavelet bases. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Philadelphia, March 2005.
- [2] A. F. Abdelnour. *Wavelet design using Gröbner bases*. PhD thesis, Polytechnic University, Brooklyn, New York, 2003.
- [3] P. Abry. *Ondelettes et Turbulences*. Diderot, Paris, 1997.
- [4] P. Abry, S. Fauve, P. Flandrin, and C. Laroche. Analysis of pressure fluctuations in swirling turbulent flows. *Journal de Physique II France*, 4:725–733, May 1994.
- [5] P. Abry and P. Flandrin. Multiresolution transient detection. In *Proc. of the IEEE-SP Int. Symp. Time-Frequency and Time-Scale Analysis*, pages 225–228, Philadelphia, October 1994.
- [6] J.-P. Antoine, R. Murenzi, and P. Vandergheynst. Directional wavelets revisited: Cauchy wavelets and symmetry detection in patterns. *Applied and Computational Harmonic Analysis*, 6(3):314–345, 1999.
- [7] J.-P. Antoine and P. Vandergheynst. 2-d Cauchy wavelets and symmetries in images. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages 597–600, Lausanne, September 16-19, 1996.
- [8] H. Ates. *Modeling location information for wavelet image coding*. PhD thesis, Princeton University, Princeton, New Jersey, 2003.
- [9] H. F. Ates and M. T. Orchard. A nonlinear image representation in wavelet domain using complex signals with single quadrant spectrum. In *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2003.
- [10] R. H. Bamberger and M. J. T. Smith. A filter bank for the directional decomposition of images: Theory and design. *IEEE Trans. on Signal Processing*, 40(4):882–893, April 1992.
- [11] B. Belzer, J.-M. Lina, and J. Villasenor. Complex, linear-phase filters for efficient image coding. *IEEE Trans. on Signal Processing*, 43(10):2425–2427, October 1995.
- [12] L. Blanc-Feraud, G. P. Bernad, and J. Zerubia. A restoration method for confocal microscopy using complex wavelet transform. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Philadelphia, March 2005.
- [13] T. J. Burns. *A non-homogeneous wavelet multiresolution analysis and its application to the analysis of motion*. PhD thesis, Air Force Institute of Technology, Ohio, 1993.

- [14] T. J. Burns, S. K. Rogers, D. W. Ruck, and M. E. Oxley. Discrete, spatiotemporal, wavelet multiresolution analysis method for computing optical flow. *Optical Engineering*, 33(7):2236–2247, July 1994.
- [15] E. J. Candés and D. L. Donoho. Curvelets, multiresolution representation, and scaling laws. In *Wavelet Applications in Signal and Image Processing VIII (Proc. SPIE)*, volume 4119, San Diego, July 2000.
- [16] E. Causevic, R. John, and J. Kovacevic. Adaptive complex wavelet-based filtering of eeg for extraction of evoked potential responses. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Philadelphia, March 2005.
- [17] W. Chan, H. Choi, and R. Baraniuk. Directional hypercomplex wavelets for multidimensional signal analysis and processing. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Montreal, May 2004.
- [18] W. Chan, H. Choi, and R. Baraniuk. Quaternion wavelets for image analysis and processing. In *Proc. IEEE Int. Conf. Image Processing*, Singapore, October 2004.
- [19] W. Chan, H. Choi, and R. G. Baraniuk. Coherent image processing using quaternion wavelets. In *Wavelet Applications in Signal and Image Processing XI*, San Diego, 2005.
- [20] J. O. Chapa and R. M. Rao. Algorithms for designing wavelets to match a specified signal. *IEEE Trans. on Signal Processing*, 48(12):3395–3406, December 2000.
- [21] C. Chaux, L. Duval, and J.-C. Pesquet. 2d dual-tree m-band wavelet decomposition. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Philadelphia, March 2005.
- [22] H. Choi, J. Romberg, R. G. Baraniuk, and N. Kingsbury. Hidden Markov tree modeling of complex wavelet transforms. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 1, pages 133–136, June 5-9, 2000.
- [23] R. R. Coifman and D. L. Donoho. Translation-invariant de-noising. In A. Antoniadis, editor, *Wavelets and Statistics*. Springer-Verlag Lecture Notes, 1995.
- [24] M. S. Crouse, R. D. Nowak, and R. G. Baraniuk. Wavelet-based signal processing using hidden Markov models. *IEEE Trans. on Signal Processing*, 46(4):886–902, April 1998.
- [25] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure Appl. Math.*, 41:909–996, 1988.
- [26] I. Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE Trans. Inform. Theory*, 36(5):961–1005, September 1990.
- [27] I. Daubechies. *Ten Lectures On Wavelets*. SIAM, 1992.
- [28] P. F. C. de Rivaz and N. G. Kingsbury. Complex wavelet features for fast texture image retrieval. In *Proc. IEEE Int. Conf. Image Processing*, Kobe, October 1999.
- [29] P. F. C. de Rivaz and N. G. Kingsbury. Bayesian image deconvolution and denoising using complex wavelets. In *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, October 8-10, 2001.
- [30] M. N. Do and M. Vetterli. Pyramidal directional filter banks and curvelets. In *Proc. IEEE Int. Conf. Image Processing*, volume 3, pages 158–161, Thessaloniki, October 2001.
- [31] M. N. Do and M. Vetterli. Contourlets: A directional multiresolution image representation. In *Proc. IEEE Int. Conf. Image Processing*, Rochester, 2002.
- [32] M. N. Do and M. Vetterli. Contourlets. In G. V. Welland, editor, *Beyond Wavelets*. Academic Press, 2003.
- [33] D. L. Donoho. Unconditional bases are optimal for data compression and for statistical estimation. *Applied and Computational Harmonic Analysis*, 1(1):100–115, December 1993.
- [34] P. L. Dragotti and M. Vetterli. Wavelet footprints: Theory, algorithms and applications. *IEEE Trans. on Signal Processing*, 51(5):1306–1323, May 2003.
- [35] J. W. Earl and N. G. Kingsbury. Spread transform watermarking for video sources. In *Proc. IEEE Int. Conf. Image Processing*, Barcelona, September 2003.

- [36] F. Fernandes, R. van Spaendonck, M. Coates, and S. Burrus. Directional complex-wavelet processing. In *Wavelet Applications in Signal and Image Processing VIII (Proc. SPIE)*, volume 4119, San Diego, July 2000.
- [37] F. Fernandes, M. Wakin, and R. G. Baraniuk. Non-redundant, linear-phase, semi-orthogonal, directional complex wavelets. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Montreal, May 2004.
- [38] F. C. A. Fernandes, I. W. Selesnick, R. L. C. van Spaendonck, and C. S. Burrus. Complex wavelet transforms with allpass filters. *Signal Processing*, 83(8):1689–1706, 2003.
- [39] F. C. A. Fernandes, R. L. C. van Spaendonck, and C. S. Burrus. A new framework for complex wavelet transforms. *IEEE Trans. on Signal Processing*, 51(7):1825–1837, July 2003.
- [40] T. C. Folsom and R. B. Pinter. Primitive features by steering, quadrature, and scale. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 20(11):1161–1173, November 1998.
- [41] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(9):891–906, September 1991.
- [42] X. Q. Gao, T. Q. Nguyen, and G. Strang. Theory and lattice structure of complex paraunitary filterbanks with filters of (hermitian-) symmetry/antisymmetry properties. *IEEE Trans. on Signal Processing*, 49(5):1028–1043, May 2001.
- [43] R. Gopinath. The phaselet transform – An integral redundancy nearly shift-invariant wavelet transform. *IEEE Trans. on Signal Processing*, 51(7):1792–1805, July 2003.
- [44] R. Gopinath. Phaselets of framelets. *IEEE Trans. on Signal Processing*, 53(5):1794–1806, May 2005.
- [45] A. Grossman and J. Morlet. Decomposition of Hardy functions into square integrable wavelets of constant shape. *SIAM J. Math. Anal.*, 15(4):723–736, July 1984.
- [46] S. Hatipoglu, S. K. Mitra, and N. G. Kingsbury. Image texture description using complex wavelet transform. In *Proc. IEEE Int. Conf. Image Processing*, Vancouver, September 2000.
- [47] M. Hemmendorff, M. T. Andersson, T. Kronander, and H. Knutsson. Phase-based multidimensional volume registration. *IEEE Trans. on Medical Imaging*, 21(12):X–X, December 2002.
- [48] P. R. Hill, D. R. Bull, and C. N. Canagarajah. Rotationally invariant texture features using the dual-tree complex wavelet transform. In *Proc. IEEE Int. Conf. Image Processing*, volume 3, pages 901–904, Vancouver, September 10-13, 2000.
- [49] P. S. Hong and M. J. T. Smith. An octave-band family of non-redundant directional filter banks. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 2, pages 1165–1168, 2002.
- [50] G. Hua. Noncoherent image denoising. Master’s thesis, Rice University, Houston, Texas, 2005.
- [51] A. Jalobeanu, N. G. Kingsbury, and J. Zerubia. Image deconvolution using hidden Markov tree modeling of complex wavelet packets. In *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, October 2001.
- [52] H. Kawabata, H. Toda, Z. Zhang, and H. Fujiwara. A new complex wavelet transform by using R-spline wavelet. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Montreal, May 2004.
- [53] N. G. Kingsbury. The dual-tree complex wavelet transform: A new efficient tool for image restoration and enhancement. In *Proc. European Signal Processing Conference*, pages 319–322, Rhodes, September 1998.
- [54] N. G. Kingsbury. The dual-tree complex wavelet transform: A new technique for shift invariance and directional filters. In *Proceedings of the Eighth IEEE DSP Workshop*, Utah, August 9-12, 1998.
- [55] N. G. Kingsbury. Image processing with complex wavelets. *Phil. Trans. Royal Society London A*, September 1999.
- [56] N. G. Kingsbury. A dual-tree complex wavelet transform with improved orthogonality and symmetry properties. In *Proc. IEEE Int. Conf. Image Processing*, Vancouver, September 10-13, 2000.
- [57] N. G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. *Applied and Computational Harmonic Analysis*, 10(3):234–253, May 2001.

- [58] N. G. Kingsbury. Design of q-shift complex wavelets for image processing using frequency domain energy minimization. In *Proc. IEEE Int. Conf. Image Processing*, Barcelona, September 2003.
- [59] N. G. Kingsbury and J. F. A. Magarey. Wavelet transforms in image processing. In *Proc. First European Conf. Signal Anal. and Prediction*, pages 23–34, Prague, June 24–27, 1997.
- [60] M. Kokare, P. K. Biswas, and B. N. Chatterji. Rotation invariant texture features using rotated complex wavelet for content based image retrieval. In *Proc. IEEE Int. Conf. Image Processing*, Singapore, October 2004.
- [61] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine. Splitting the unit delay. *IEEE Signal Processing Magazine*, 13(1):30–60, January 1996.
- [62] M. Lang. Allpass filter design and applications. *IEEE Trans. on Signal Processing*, 46(9):2505–2514, September 1998.
- [63] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells, Jr. Noise reduction using an undecimated discrete wavelet transform. *IEEE Signal Processing Letters*, 3(1):10–12, January 1996.
- [64] J.-M. Lina and M. Mayrand. Complex Daubechies wavelets. *Applied and Computational Harmonic Analysis*, 2:219–229, 1995.
- [65] E. Lo, M. Pickering, M. Frater, and J. Arnold. Scale and rotation invariant texture features from the dual-tree complex wavelet transform. In *Proc. IEEE Int. Conf. Image Processing*, Singapore, October 2004.
- [66] P. Loo and N. G. Kingsbury. Digital watermarking using complex wavelets. In *Proc. IEEE Int. Conf. Image Processing*, Vancouver, September 2000.
- [67] J. F. A. Magarey and N. G. Kingsbury. Motion estimation using a complex-valued wavelet transform. *IEEE Trans. on Signal Processing*, 46(4):1069–84, April 1998.
- [68] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Patt. Recog. and Mach. Intell.*, 11(7):674–693, July 1989.
- [69] S. Mallat. *A wavelet tour of signal processing*. Academic Press, 1998.
- [70] H. Malvar. A modulated complex lapped transform and its applications to audio processing. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Phoenix, March 16–19, 1999.
- [71] H. Malvar. Fast algorithm for the modulated complex lapped transform. *IEEE Signal Processing Letters*, 10(1):8–10, January 2003.
- [72] M. K. Mihcak, I. Kozintsev, K. Ramchandran, and P. Moulin. Low-complexity image denoising based on statistical modeling of wavelet coefficients. *IEEE Signal Processing Letters*, 6(12):300–303, December 1999.
- [73] M. Miller, N. Kingsbury, and R. Hobbs. Seismic imaging using complex wavelets. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Philadelphia, March 2005.
- [74] F. Mintzer. Filters for distortion-free two-band multirate filter banks. *IEEE Trans. on Acoust., Speech, Signal Proc.*, 33(3):626–630, June 1985.
- [75] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.
- [76] H. Ozkaramanli and R. Yu. On the phase condition and its solution for Hilbert transform pairs of wavelet bases. *IEEE Trans. on Signal Processing*, 51(12):3293–3294, December 2003.
- [77] A. Papoulis. *Signal Analysis*. McGraw-Hill, 1977.
- [78] J. Portilla, V. Strela, M. Wainwright, and E. P. Simoncelli. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. on Image Processing*, 12(11):1338–1351, November 2003.
- [79] T. H. Reeves and N. G. Kingsbury. Overcomplete image coding using iterative projection-based noise shaping. In *Proc. IEEE Int. Conf. Image Processing*, Rochester, September 2002.
- [80] J. Romberg, H. Choi, R. G. Baraniuk, and N. G. Kingsbury. Multiscale classification using complex wavelets and hidden Markov tree models. In *Proc. IEEE Int. Conf. Image Processing*, volume 2, pages 371–374, Vancouver, September 10–13, 2000.

- [81] J. Romberg, M. Wakin, H. Choi, and R. G. Baraniuk. A geometric hidden Markov tree wavelet model. In *Wavelet Applications in Signal and Image Processing X (Proc. SPIE 5207)*, San Diego, 2003.
- [82] J. K. Romberg, H. Choi, and R. G. Baraniuk. Multiscale edge grammars for complex wavelet transforms. In *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, October 2001.
- [83] J. K. Romberg, M. Wakin, H. Choi, N. G. Kingsbury, and R. G. Baraniuk. A hidden Markov tree model for the complex wavelet transform. *Rice ECE Technical Report*, September 2002.
- [84] J. G. Rosiles and M. J. T. Smith. A low complexity overcomplete directional image pyramid. In *Proc. IEEE Int. Conf. Image Processing*, Barcelona, September 2003.
- [85] H. W. Schüßler and P. Steffen. On the design of allpasses with prescribed group delay. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 3, pages 1313–1316, Albuquerque, April 3-6, 1990.
- [86] I. W. Selesnick. The double density DWT. In A. Petrosian and F. G. Meyer, editors, *Wavelets in Signal and Image Analysis: From Theory to Practice*. Kluwer, 2001.
- [87] I. W. Selesnick. Hilbert transform pairs of wavelet bases. *IEEE Signal Processing Letters*, 8(6):170–173, June 2001.
- [88] I. W. Selesnick. The design of approximate Hilbert transform pairs of wavelet bases. *IEEE Trans. on Signal Processing*, 50(5):1144–1152, May 2002.
- [89] I. W. Selesnick. The double-density dual-tree discrete wavelet transform. *IEEE Trans. on Signal Processing*, 52(5):1304–1314, May 2004.
- [90] I. W. Selesnick and K.-L. Li. Video denoising using 2D and 3D dual-tree complex wavelet transforms. In *Wavelet Applications in Signal and Image Processing X (Proc. SPIE 5207)*, San Diego, 2003.
- [91] L. Sendur and I. W. Selesnick. Bivariate shrinkage with local variance estimation. *IEEE Signal Processing Letters*, 9(12):438–441, December 2002.
- [92] C. W. Shaffrey, N. G. Kingsbury, and I. H. Jermyn. Unsupervised image segmentation via Markov trees and complex wavelets. In *Proc. IEEE Int. Conf. Image Processing*, Rochester, 2002.
- [93] F. Shi and I. W. Selesnick. Video denoising using oriented complex wavelet transforms. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, June 2004.
- [94] F. Shi, I. W. Selesnick, and S. Cai. Image sharpening via image denoising in the complex wavelet domain. In *Wavelet Applications in Signal and Image Processing X (Proc. SPIE 5207)*, San Diego, 2003.
- [95] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, October 1995.
- [96] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multi-scale transforms. *IEEE Trans. Inform. Theory*, 38(2):587–607, March 1992.
- [97] K. Sivaramakrishnan and T. Nguyen. A uniform transform domain video codec based on dual tree complex wavelet transform. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 3, pages 1821–1824, May 7-11, 2001.
- [98] A. Skodras, C. Christopoulos, and T. Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, September 2001.
- [99] M. J. T. Smith and T. P. Barnwell III. Exact reconstruction for tree-structured subband coders. *IEEE Trans. on Acoust., Speech, Signal Proc.*, 34(3):431–441, June 1986.
- [100] J.-L. Starck, E. J. Candés, and D. L. Donoho. The curvelet transform for image denoising. *IEEE Trans. on Image Processing*, 11(6):670–684, June 2000.
- [101] J.-L. Starck, M. Elad, and D. L. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132, 2004.
- [102] J.-L. Starck and F. Murtagh. Astronomical image and signal processing: looking at noise, information and scale. *IEEE Signal Processing Magazine*, 18(2):30–40, March 2001.
- [103] G. Strang. Wavelets and dilation equations: A brief introduction. *SIAM Review*, 31:614–627, 1989.

- [104] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.
- [105] D. B. H. Tay and M. Palaniswami. Design of approximate Hilbert transform pair of wavelets with exact symmetry. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Montreal, May 2004.
- [106] J. P. Thiran. Recursive digital filters with maximally flat group delay. *IEEE Trans. on Circuit Theory*, 18(6):659–664, November 1971.
- [107] P. P. Vaidyanathan. *Multirate Systems and Filter Banks*. Prentice Hall, 1993.
- [108] R. van Spaendonck, T. Blu, R. Baraniuk, and M. Vetterli. Orthogonal Hilbert transform filter banks and wavelets. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, volume 6, pages 505–508, April 6–10, 2003.
- [109] R. L. C. van Spaendonck. *Seismic Applications of Complex Wavelet Transforms*. PhD thesis, Delft University of Technology, Delft, The Netherlands, 2002.
- [110] P. Vandergheynst and J.-F. Gobbers. Directional dyadic wavelet transforms: design and algorithms. *IEEE Trans. on Image Processing*, 11(4):363–372, April 2002.
- [111] M. Vetterli. Filter banks allowing perfect reconstruction. *Signal Processing*, 10(3):219–244, 1986.
- [112] M. Vetterli. Wavelets, approximation, and compression. *IEEE Signal Processing Magazine*, 18(5):59–73, September 2001.
- [113] M. Wakin. Image compression using multiscale geometric edge models. Master’s thesis, Rice University, Houston, Texas, 2002.
- [114] M. Wakin, M. Orchard, R. G. Baraniuk, and V. Chandrasekaran. Phase and magnitude perceptual sensitivities in nonredundant complex wavelet representations. In *Proc. Asilomar Conf. Signals, Systems, and Computers*, 2003.
- [115] B. Wang, Y. Wang, I. Selesnick, and A. Vetro. Video coding using 3-D dual-tree discrete wavelet transforms. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Philadelphia, March 2005.
- [116] Z. Wang and E. Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, Philadelphia, March 2005.
- [117] Z. Wang and E. P. Simoncelli. Local phase coherence and the perception of blur. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Adv. Neural Information Processing Systems*, volume 16. MIT Press, May 2004.
- [118] Z. Ye and C.-C. Lu. A complex wavelet domain Markov model for image denoising. In *Proc. IEEE Int. Conf. Image Processing*, volume 3, pages 365–368, Barcelona, September 2003.
- [119] R. W. Young and N. G. Kingsbury. Motion estimation using lapped transforms. In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing (ICASSP)*, San Francisco, March 1992.
- [120] R. W. Young and N. G. Kingsbury. Video compression using lapped transforms for motion estimation/compensation and coding. *Optical Engineering*, 32(7):1451–1463, July 1993.
- [121] R. Yu and H. Ozkaramanli. Hilbert transform pairs of biorthogonal wavelet bases. *IEEE Trans. on Signal Processing*, 2005. To appear.
- [122] R. Yu and H. Ozkaramanli. Hilbert transform pairs of orthogonal wavelet bases: necessary and sufficient conditions. *IEEE Trans. on Signal Processing*, 2005. To appear.
- [123] X.-P. Zhang, M. D. Desai, and Y.-N. Peng. Orthogonal complex filter banks and wavelets: Some properties and design. *IEEE Trans. on Signal Processing*, 47(4):1039–1048, April 1999.