Reproducibility Standards for Wavelet Transform Algorithms

Carl Taswell

C. Taswell is with UCSD School of Medicine, La Jolla, CA 92093-0603, (ctaswell@ucsd.edu), and with Computational Toolsmiths, POB 9925, Stanford, CA 94309-9925, (taswell@toolsmiths.com). Manuscript submitted 12/15/96 to IEEE TCAS-II; revision resubmitted 3/12/98.

Abstract

Specifications for reproducibility standards are developed for wavelet transform algorithms. Reproducibility of an algorithm is defined as the requirement that the specified algorithm yield the same transform coefficients for the same signal data regardless of implementation in any programming language or on any computing machine.

The specification is built with three heirarchical stages consisting of 1) filter bank coefficients, 2) single-level convolutions, and 3) multi-level transforms. Each stage is specified with all the necessary choices, parameters, and operators required to insure reproducibility. Each stage can be further characterized by additional properties that provide relevant information. The specification has been designed to be a sufficiently general and flexible framework which encompasses many different convolution types addressing the issues of both the phase shift and the boundary treatment.

New convolution phase shift variants are introduced. In particular, a peak near-aligned phase variant is demonstrated on test signals and applied to fast wavelet based matrix multiplication. Finally, in the context of computational science and engineering, the concept of scientific reproducibility of an algorithm is discussed and contrasted with two other concepts introduced as repetitive executability and input-output repeatability.

Keywords

Wavelet transforms, multirate filter banks, algorithms, repetitive executability, input-output repeatability, scientific reproducibility, standards.

I. Introduction

Growth in the number of variations and applications of wavelet transforms has progressed rapidly over the past decade [1]. Indeed, this growth has been so expansive that there are now many different classes and subclasses of wavelets and wavelet transforms. Certainly, opinions and definitions of what constitutes a wavelet and wavelet transform vary with different authors [2]. Nevertheless, this report focuses on those wavelet transforms that are currently the most prevalent and that can be implemented as iterated filter banks. Although particular attention is addressed to non-redundant transforms, the general principles of the methods described here can also be applied to redundant transforms.

Most of the literature on wavelet transforms has discussed the theory of analysis and methods rather than the implementation of algorithms. There have been a few important and notable exceptions such as the papers by Shensa [3] and Rioul and Duhamel [4]. However, these articles discussed algorithmic schemes at a more general level in order to describe them and compare their relative efficiency, rather than algorithmic implementations at a sufficiently detailed level to specify them and insure their reproducibility.

The use of the terms *describe* and *description* when associated here with the discussion of an algorithm will refer to a general scheme or diagram for the algorithm, whereas the terms *specify* and *specification* will refer to a complete listing of all implementation details for the algorithm. With this usage then, a description of an algorithm is sufficient for a discussion of its *efficiency*, but a specification is necessary for a discussion of its *reproducibility*.

A complete specification of an algorithm may be provided with a detailed pseudo-code template as exemplified in the wavelet transform algorithm published by Taswell and McGill [5] or with a sufficiently detailed listing of all mathematical equations and parameters as exemplified by the work of Bradley and Brislawn [6] for the FBI fingerprint image compression standard. However, these published examples remain more the exception than the rule. Systematic development of a standard for the specification and reproducibility of wavelet transform algorithms has not yet been promoted in the wavelet community.

Thus, this report presents the first attempt in the field of wavelets to develop a systematic methodology to specify mathematically, and then to evaluate numerically wavelet filter banks, convolutions, and transform algorithms in a heirarchical framework with empirical testing and validation of each stage. Any such methodology fulfilling this objective will constitute an important and necessary aspect of reporting computational experiments involving wavelet transforms. These algorithm verification methodologies can then be used to insure reproducibility of results, especially for those experimental studies purporting to compare alternative algorithms.

The evaluation methodology presented here comprises a systematic listing of the principal parameters, choices, and tests that can be specified and performed for wavelet filter coefficients, single-level convolutions, and multi-level transforms when the investigator wishes to guarantee reproducibility and verifiability regardless of computing platform and programming language. The specification of the filter convolutions, the phase delays and advances of the filters in the filter bank, and the treatment of the ends of the signal remains a central issue relevant to algorithms for finite-length signals. In the introduction to his paper [7], Brislawn provides a comprehensive historical review of the various convolution types available. However, reporting of such details is often neglected.

To emphasize the importance of specifying these convolution details, this article presents a unifying framework for reporting them and demonstrates the use of this framework with a simple yet novel solution to the phase alignment problem. As an example application, this solution is then applied to fast wavelet based matrix multiplication. Beylkin et al [8] introduced fast algorithms for the efficient multiplication of sparse wavelet based representations for integral and pseudodifferential matrix operators of a certain class. Keinert [9] continued this work by implementing the Beylkin algorithm for biorthogonal instead of orthogonal wavelets, and observing the relative advantages and disadvantages of the various wavelets investigated. In the example application demonstrated in Section III-D, their work is extended further by experimentally comparing different convolution phase variants instead of different wavelets. Finally, Section IV discusses how the specification and evaluation methodology reported here promotes *scientific reproducibility* in contrast with *repetitive executability*, which is a term coined here to refer to the notions advocated by several other authors [10], [11], [12]. Earlier versions of material reported in this article have appeared elsewhere [13], [14].

II. Methods

Algorithms are specified here by building heirarchical systems with modular components consisting of structures and functions for filter banks, single-level convolutions, and multi-level transforms. Each component, whether a structure containing data or a function processing data, is represented as a data matrix or an operator matrix. Each stage of the heirarchy is detailed with all necessary choices to insure reproducibility and verifiability. The specification outlined here assumes that the multi-level wavelet transform can be implemented as an iteration of a multi-rate single-level convolution of the filters in an M-band filter bank. Complete algorithmic details for all of the methods presented here have been and/or will be available in a sequence of papers including [15], [16], a forthcoming book [17], and a software function library [18], [19].

A. Filter Banks

Consider an M-band analysis and synthesis filter bank system with uniform downsampling and upsampling rate R. This system has M analysis filters with impulse responses $\mathbf{f}_m \equiv f_m(n)$, M downsamplers and upsamplers operating at rate R, and M synthesis filters $\mathbf{g}_m \equiv g_m(n)$ where $m = 0, 1, \ldots, M-1$ is the band index and $n = 0, 1, \ldots, N-1$ is the time index. Here $N = QR$ is an integer multiple $Q = \lceil (\max_m N_m)/R \rceil$ of R determined with the maximum of the minimum support lengths N_m of $\mathbf{f}_m$.[1] The first nonzero coefficient of each $\mathbf{f}_m$ is indexed at time step $n = 0$ and any filter with length $N_m < N$ is padded with trailing zeros. The first nonzero coefficient of each $\mathbf{g}_m$ is indexed at a time step $n \geq 0$ and padded with either leading or trailing zeros or both as long as the total length with padding is constrained to N.

The filter coefficients can then be structured as the matrices $\mathbf{F} = [f_{nm}]$ and $\mathbf{G} = [g_{nm}]$ with time index n increasing down the rows and band index m increasing across the columns. This convention permits columnwise tabulation of the coefficients and facilitates convenient columnwise analysis for the band filters in each of the columns. Thus, individual filters in the filter banks can be readily characterized by computing various measures of each column of coefficients in the matrices.

A minimal specification of the filter bank coefficients requires either a) actual tabulation of the coefficient matrices $\mathbf{F}$ and $\mathbf{G}$, or b) specific definition of the computational algorithm that generates the coefficient matrices with sufficient detail to clarify choices of signs, phases, and normalization constants. Assuming that $\mathbf{F}$ and $\mathbf{G}$ have been unequivocally specified, additional informative characterization of the filter banks may also include a) the accuracy and precision of the numerical coefficients relative to their theoretical values, b) various norms and statistical moments, the number v of vanishing moments, the Holder estimate h for time-domain regularity, estimates of the frequency-domain selectivity, and other measures of the individual filters in the filter banks, c) the system delay Δ and reconstruction error ϵ for an impulse processed through the filter bank system, and d) other properties of the filter bank system rather than the individual filters.

The delay Δ and error ϵ can be computed most readily with a simple modification of the method devised by Nayebi *et al.* [20]. The Holder regularity h can be estimated by the method of Rioul [21] or Taswell [15]. The number v of vanishing moments can be numerically tested by straightforward calculations subject to a prescribed error tolerance. Such a definition interprets “vanishing” to mean that the required “zero” is any absolute numerical value less than the prescribed error tolerance. Finally, the filters may also be tested numerically for other properties such as orthogonality and biorthogonality. All of these methods and tests are detailed in [15], [16].

[1] The minimum support length of a filter is its length without any leading or trailing zeros.

B. Single-Level Convolutions

Under the assumptions validating the noble identities, the order of analysis filters and downsamplers can be exchanged, and similarly the order of upsamplers and synthesis filters can be exchanged [22]. Moreover, for computational efficiency, each pair of operations can be integrated into a single convolution operation called downscaling for the composition of analysis filtering and downsampling, and upscaling for the composition of upsampling and synthesis filtering [21]. Thus, for the purposes of this exposition, the operations will be denoted with the matrices $\mathbf{T}(\mathbf{f}_m)$ for filtering with the m^{th} analysis filter $\mathbf{f}_m$, $\mathbf{D}$ for downsampling, $\mathbf{D}_m \equiv \mathbf{D} \cdot \mathbf{T}(\mathbf{f}_m)$ for downscaling with $\mathbf{f}_m$, $\mathbf{U}$ for upsampling, $\mathbf{T}(\mathbf{g}_m)$ for filtering with the m^{th} synthesis filter $\mathbf{g}_m$, and $\mathbf{U}_m \equiv \mathbf{T}(\mathbf{g}_m) \cdot \mathbf{U}$ for upscaling with $\mathbf{g}_m$. Since the matrix operators $\mathbf{T}$ are assumed here to implement standard linear or circular convolution (denoted respectively $\mathbf{T}^{\text{lin}}$ or $\mathbf{T}^{\text{cir}}$), they are banded Toeplitz matrices and thus the matrices $\mathbf{D}_m$ and $\mathbf{U}_m$ are block Toeplitz matrices.

Using this matrix notation, the single-level convolutions can be implemented and studied as multiplications of the finite-length signal data $\mathbf{X}$ with the finite-size downscaling matrices $\mathbf{D}_m$ to obtain the decomposition bands $\mathbf{Y}_m = \mathbf{D}_m\mathbf{X}$, and then with the upscaling matrices $\mathbf{U}_m$ to obtain the reconstruction bands $\hat{\mathbf{X}}_m = \mathbf{U}_m\mathbf{Y}_m$. Summing these outputs yields the final reconstruction $\hat{\mathbf{X}} = \sum_m \hat{\mathbf{X}}_m$. In this representation, the matrices $\mathbf{X}$ and $\mathbf{Y}$ could be replaced by the vectors $\mathbf{x}$ and $\mathbf{y}$. These alternatives correspond to single- and multi-channel signal data with single and multiple columns for the vectors and matrices, respectively. Common examples of multi-channel data are two-channel (left and right stereo) audio recordings and twelve-channel electrocardiograms.

Discussion of the matrix representation of the single-level convolutions suffices to fix issues related to reproducibility without concern for efficiency. Again, this objective is defined here principally as the requirement that a given sequence of output coefficients be computed reproducibly for a given sequence of input coefficients. Thus, issues related to efficiency (such as matrix-filter versus vector-filter implementations [5], standard filter versus lattice filter implementations [22], and time-domain versus frequency-domain implementations [4]) are not considered here other than as already mentioned at the beginning of this section.

It is, however, the finite size of the downscaling, upscaling, and data matrices that does directly impact reproducibility of the single-level convolutions, and consequently, the multi-level transforms. This finiteness imposes the necessity to consider the treatment of the ends of the signal, not only with regard to the choice of the type of convolution such as zero-extended [5], circularly-periodized [5], linearly-extended [23], [24], symmetrically-reflected [7], or boundary-adjusted [25], but also with regard to the choice of phase shifts for the convolutions. To specify the single-level convolutions reproducibly, it is thus necessary to clarify unambiguously the convolution types and phase delays and advances imposed on the filter bands in the filter banks. Clarifying in terms of the heirarchy of filter banks, single-level convolutions, and multi-level transforms, the matrices $\mathbf{F}$ and $\mathbf{G}$ determine a standardized M-band filter bank system with all filters in the causal time-aligned zero-indexed format as explained in Section II-A, while the matrices $\{\mathbf{D}_m, \mathbf{U}_m | m = 0, \ldots, M-1\}$ determine an unrestricted, possibly anti-causal, M-band single-level convolution system for a decomposition and reconstruction allowing for many possible variations of signal-end treatment.

All of the different types of convolutions can be incorporated in the following general framework described here with analysis phase delays α_{im}, synthesis phase delays β_{im}, and several additional matrix operators: the pre-processing or extension matrix $\mathbf{E}$, the shift matrix $\mathbf{S}$, and the post-processing or restriction matrix $\mathbf{R}$. Then the m^{th} analysis downscaling and synthesis upscaling matrices can be redefined as

$$\begin{aligned}\mathbf{D}_m &\equiv \mathbf{R}\cdot\mathbf{D}\cdot\mathbf{T}(\mathbf{f}_m, \alpha_{2m})\cdot\mathbf{E}(\alpha_{1m})\\ \mathbf{U}_m &\equiv \mathbf{R}\cdot\mathbf{S}(\beta_{3m})\cdot\mathbf{T}(\mathbf{g}_m, \beta_{2m})\cdot\mathbf{U}\cdot\mathbf{E}(\beta_{1m})\end{aligned}$$

for a scheme intended to impose a perfect reconstruction result $\mathbf{I} = \sum_m \mathbf{U}_m \cdot \mathbf{D}_m$ on a single-level decomposition and reconstruction whenever possible. Note that $\mathbf{S}(\beta_{3m})$ is a final shift necessary to account for the combined delays resulting from the operators $\mathbf{E}$ and $\mathbf{T}$ as well as from the delay Δ for the filter banks $\mathbf{F}$ and $\mathbf{G}$ in their standardized format. This scheme assumes zero delays on the $\mathbf{D}$, $\mathbf{U}$, and $\mathbf{R}$ operators but still allows for as many as $5M$ different delay parameters for the $\mathbf{E}$, $\mathbf{T}$, and $\mathbf{S}$ operators used here in the M-band single-level convolution system.

Phase alignment of peaks of polyphase components of bands in the transform domain relative to the signal domain can be accomplished by the simple introduction of two more circular shift operators and delay parameters in the scheme

$$\begin{aligned}\mathbf{D}_m &\equiv \mathbf{S}(\alpha_{3m})\mathbf{R}\mathbf{D}\mathbf{T}(\mathbf{f}_m, \alpha_{2m})\mathbf{E}(\alpha_{1m})\\ \mathbf{U}_m &\equiv \mathbf{R}\mathbf{S}(\beta_{4m})\mathbf{T}(\mathbf{g}_m, \beta_{3m})\mathbf{U}\mathbf{E}(\beta_{2m})\mathbf{S}(\beta_{1m})\end{aligned}$$

which ideally should require that the final downscaling rotation $\mathbf{S}(\alpha_{3m})$ and the initial upscaling inverse rotation $\mathbf{S}(\beta_{1m})$ yield the identity

$$\mathbf{I} = \mathbf{S}(\beta_{1m}) \cdot \mathbf{S}(\alpha_{3m}).$$

Thus, imposing $\beta_{1m} = -\alpha_{3m}$ eliminates M of the additional parameters, and absorbing $\mathbf{S}(\beta_{1m})$ into $\mathbf{E}(\beta_{2m})$ eliminates another M of the additional operators. Relabeling indices such that the main Toeplitz operators $\mathbf{T}(\mathbf{f})$ and $\mathbf{T}(\mathbf{g})$ are assigned delay index $i = 1$, the inner operators $\mathbf{S}$ and $\mathbf{E}$ are assigned $i = 2$, and the outer operators $\mathbf{E}$ and $\mathbf{S}$ are assigned $i = 3$ yields the scheme

$$\begin{aligned} \mathbf{D}_m &\equiv \mathbf{S}(\alpha_{2m})\mathbf{RDT}(\mathbf{h}_m, \alpha_{1m})\mathbf{E}(\alpha_{3m}) \\ \mathbf{U}_m &\equiv \mathbf{RS}(\beta_{3m})\mathbf{T}(\mathbf{g}_m, \beta_{1m})\mathbf{UE}(\beta_{2m}) \end{aligned}$$

as a general framework sufficient to account for the various convolution types. This particular indexing convention was adopted for WAVBOX 4.4 Software [18] used to produce the results reported in Section III. Detailed algorithms including pseudo-code templates for these phase aligned convolution types will be available elsewhere [17].

A brief explanation, however, is provided here. Let μ_{rm} (for $r = 0, \dots, R-1$ and $m = 0, \dots, M-1$) be the indices of the center peaks of the r^{th} polyphase components of the m^{th} band filters of the analysis filter bank $\mathbf{F}$. Let μ_{Rm} be the indices of the center peaks of the entire m^{th} band filters of the analysis filter bank $\mathbf{F}$. Then one possible phase near-aligned solution involves: 1) choosing $\mu^\star = \mu_{rm}$ for a particular r and m, 2) setting all α_{1m} to the delay ζ computed as a function of R and $\mu^\star$, 3) setting all α_{3m} to the delay η computed as a function of R and N, 4) setting each α_{2m} to a delay computed as a function of R and the sum $\mu_{rm} + \zeta + \eta$, 5) setting $\beta_{2m} = -\alpha_{2m}$, and 6) setting $\beta_{3m} = -\Delta - \zeta - \eta$. Numerous other solutions are possible depending on a) the definitions assumed for the center peaks of polyphase components or entire filters, b) the definitions assumed for phase alignment or near-alignment, and c) any other constraints imposed on the problem.

Thus, a minimal specification for reproducibility of the single-level convolutions requires a) the convolution type including the composition sequence of the various operators as in the general framework above, b) the algorithms for generating the operators with particular attention to the extension operator $\mathbf{E}$ for a given extension or boundary treatment type, c) any auxiliary parameters or boundary filters necessary for $\mathbf{E}$, d) restriction length parameters necessary for the restriction operator $\mathbf{R}$, and e) the phase shifts necessary for any of the $\mathbf{E}$, $\mathbf{T}$, and $\mathbf{S}$ operators used by the convolution type (or the algorithms for setting the phase shifts). Additional characterization for verifiability of the single-level convolutions may also include: a) comparison of results with known sequences of transform-domain decomposition coefficients for given sequences of signal-domain test data, b) the reconstruction error ε for the test signals resulting from use of the convolutions as a single-level decomposition and reconstruction, and c) various other measures designed to reveal properties of the convolution type such as energy conservation, distribution, and shift.

Simple definitions are possible for the latter measures as ratios. Let the energy conservation ratio ρ_c be the ratio of the energy of $\{\mathbf{Y}_m | m = 0, \ldots , M-1\}$ to the energy of $\mathbf{X}$. Let the energy distribution ratio ρ_d be the ratio of the energy of $\{\mathbf{Y}_m | m = 1, \ldots , M-1\}$ to the energy of $\mathbf{Y}_0$. In order to define the energy shift ratio ρ_s and also enable visualization of various aspects of the convolution including the behavior of the polyphase components in response to the M-band filters and the boundary treatment, a simple test signal called a "multiple M-spike" has been designed. This test signal has M-channels in which each channel has impulses near the beginning, distributed through the middle, and at the end of the signal, but the impulses for each channel are shifted relative to each other by one time index. Thus, each channel is intended to test a different polyphase component. In conjunction with this test signal, the energy shift ratio ρ_s has been defined to track the energy displaced by the phase shifts of the convolution. This measure, computed by tracking energy in blocks of length R for all M channels of the M-spike test signal, has values in the range $0 \leq \rho_s \leq 1$ with a value of $\rho_s = 0$ indicating that no energy has been displaced by more than R time units. Note however that the impulses in each channel of the M-spike test signal must be spaced more than R time units apart in order for this measure to be meaningful.

C. Multi-Level Transforms

Given analysis and synthesis filter bank coefficients specified by $\mathbf{F}$ and $\mathbf{G}$ (Section II-A) used to construct downscaling and upscaling single-level convolution operators specified by $\mathbf{D}_m$ and $\mathbf{U}_m$ (Section II-B), then a multi-level transform algorithm can be specified as the procedure by which $\mathbf{D}_m$ and $\mathbf{U}_m$ (sized appropriately for each level l) are used iteratively to process the input signal and compute the output transform. For an L-level M-band wavelet transform which iterates on the lowpass filter band indexed $m = 0$, a pseudo-code template for the forward transform algorithm can be written as

```
$\mathbf{Y}_0^0 = \mathbf{X}$
for $l = 0 : L-1$
    for $m = 0 : M-1$
        $\mathbf{Y}_m^{l+1} = \mathbf{D}_m^l \mathbf{Y}_0^l$
    end
end
```

and for the inverse wavelet transform algorithm as

```
$\hat{\mathbf{X}}_0^L = \mathbf{Y}_0^L$
for $l = L : -1 : 1$
    $\hat{\mathbf{X}}_0^{l-1} = \mathbf{U}_0^l \hat{\mathbf{X}}_0^l + \sum_1^{M-1} \mathbf{U}_m^l \mathbf{Y}_m^l$
end
$\hat{\mathbf{X}} = \hat{\mathbf{X}}_0^0$
```

with specific algorithms requiring definition of the object structures used for storage of the coefficients (or alternatively, the sequence of coefficients in an output file) in a manner analogous to the example published in ACM TOMS Algorithm 735 [5].

Thus, a minimal specification for reproducibility of a multi-level transform algorithm requires a) the filter bank coefficients $\mathbf{F}$ and $\mathbf{G}$, b) the single-level convolution operators $\mathbf{D}_m^l$ and $\mathbf{U}_m^l$, c) the algorithmic scheme by which the convolution operators are iterated, d) the parameter L for the number of levels of iteration, and e) the transform coefficient object structures with locations of coefficients in the object structures or file output sequences. Additional characterization for verifiability of the multi-level transform algorithm may also include: a) known sequences of transform coefficients for given sequences of test signal coefficients, and b) the reconstruction error $\mathcal{E}$ for the test signals under various norms and conditions. For example, degradation of the signal can be tracked through multiple cycles of decomposition and reconstruction:

```
$\hat{\mathbf{X}} = \mathbf{X}$
for $k = 1 : K$
    $\mathbf{Y} = \mathrm{fwt}(\hat{\mathbf{X}})$
    $\hat{\mathbf{X}} = \mathrm{iwt}(\mathbf{Y})$
    $\mathcal{E}(k) = \mathrm{wtre}(\mathbf{X}, \hat{\mathbf{X}})$
end
```

where the function `fwt` is the forward wavelet transform, `iwt` the inverse wavelet transform, and `wtre` the wavelet transform reconstruction error. Plots of $\mathcal{E}(k)$ versus k can be used to obtain empirical estimates of the error growth rates as a function of the cycle k.

D. Error Types

Reporting any error value also requires that the type of error be specified. Common error types include those defined by the ℓ^p vector norms. In addition, let the following elementwise error types be defined for an arbitrary matrix $\mathbf{X}$ and its estimate $\hat{\mathbf{X}}$ with respect to the matrix elements x_{ij} and $\hat{x}_{ij}$: the maximum absolute value error

$$\mathrm{mav}(\mathbf{X}, \hat{\mathbf{X}}) = \max_{i,j} |x_{ij} - \hat{x}_{ij}|,$$

the maximum relative value error

$$\mathrm{mrv}(\mathbf{X}, \hat{\mathbf{X}}) = \max_{i,j} |(x_{ij} - \hat{x}_{ij})/x_{ij}|,$$

and the maximum mixed value error

$$\mathrm{mmv}(\mathbf{X}, \hat{\mathbf{X}}) = \max_{i,j} |x_{ij} - \hat{x}_{ij}|/(1 + |x_{ij}|).$$

All errors reported in Section III-A are maximum absolute value errors. Errors reported in Sections III-B, III-C, and III-D are maximum mixed value errors unless noted otherwise.

E. Software and Hardware

Numerical and graphical results reported here were computed with Version 4.4a3 (29-Dec-96) of WAVBOX Software [18], [19], [17] running under Version 4.2c.1 (3-Oct-94) of the MATLAB technical computing environment [26] on a Toshiba Tecra 720CDT with a 133 MHz Pentium and the Windows 95 operating system.

III. Results

All computations were performed for a critically sampled wavelet transform multirate filter bank system with $R = M = 2$. In analogy with the use of the term wavelet to refer to filters and functions corresponding to the highpass band, the term scalet will be used to refer to filters and functions corresponding to the lowpass band.

A. Filter Banks

The Daubechies' compact orthogonal least-asymmetric filters [27] of length $N = 8$ were generated with the algorithm described by Taswell [28], [29], [15]. Sign, phase, and norm were chosen such that the coefficient $f_{00} = -7.577 \times 10^{-2}$. Coefficients of all filters in the filter bank were normalized in the ℓ^2-norm to one. Tests of the scalets (lowpass filters $\mathbf{f}_0$ and $\mathbf{g}_0$) yielded results of $v = 0$ vanishing moments and $h = 1.403$ Holder regularity estimates. The wavelets (highpass filters $\mathbf{f}_1$ and $\mathbf{g}_1$) were also tested and confirmed to have $v = 4$ vanishing moments with an error of 8.49×10^{-12}. All of the filters were confirmed to be orthogonal with an error of 4.22×10^{-13}. Values of $\Delta = 7$ and $\epsilon = 4.22 \times 10^{-13}$ for the filter banks $\mathbf{F}$ and $\mathbf{G}$ were obtained with the modified Nayebi-Barnwell-Smith perfect reconstruction test [20], [16].

B. Single-Level Convolutions

A circularly-periodized convolution type was chosen for the single-level decomposition and reconstruction steps and was tested with a quadruple M-spike M-channel test signal with $M = 2$. Figure 1 displays two different phase variants of this convolution type: a causal analysis variant called peak non-aligned with phase delays $\alpha = [0, 0; 0, 0; 0, 0]$ and $\beta = [0, 0; 0, 0; -7, -7]$, and an anti-causal analysis variant called peak near-aligned with phase delays $\alpha = [1, 1; -2, -3; 0, 0]$ and $\beta = [0, 0; 2, 3; -8, -8]$. The reconstruction error was $\varepsilon = 2.96 \times 10^{-13}$ for both phase variants. The energy conservation and distribution ratios were $\rho_c = 1.000$ and $\rho_d = 1.000$ for both phase variants. The energy shift ratio was $\rho_s = 0.954$ and $\rho_s = 0.084$ for the peak non-aligned and near-aligned variants, respectively.

C. Multi-Level Transforms

Using the configurations as described in Sections III-A and III-B, the single-level steps were iterated to $L = 5$ levels on a single-channel test signal called "peaks & chasms" with length 512 samples. Figure 2 displays the approximations output by the scalets and details output by the wavelets for each of the levels and each of the phase variants. In the usual "discrete wavelet transform" or "fast wavelet transform", only the wavelet details from levels $l = 1, \ldots, 5$ and the scalet approximation from level $l = 5$ would be retained for storage or further processing as the non-redundant transform.

The single-level steps were also iterated to $L = 5$ levels, and tested for $K = 100$ cycles of forward and inverse transforms on a single-channel test signal called "random normal" with length 512 samples. Figure 3 displays log-log plots of the reconstruction error $\mathcal{E}(k)$ as a function of k. Linear regression estimates of the slopes of the error curves for each of the ℓ^1, ℓ^2, and ℓ^∞ error norms resulted in values of 1.00 yielding the empirical observation

$$\log_{10} \mathcal{E}(k) = \mathcal{E}(1) + \log_{10} k.$$

Values of $\mathcal{E}^1(1) = 1.34 \times 10^{-10}$, $\mathcal{E}^2(1) = 7.36 \times 10^{-12}$, and $\mathcal{E}^\infty(1) = 9.56 \times 10^{-13}$ were obtained for the ℓ^1, ℓ^2, and ℓ^∞ error norms, respectively.

D. Application to Fast Matrix Multiplication

Again using the configurations as described in Sections III-A and III-B, the single-level steps were iterated to $L = 4$ levels in a separable 2-D wavelet transform for an application to fast-wavelet-based matrix multiplication as described by Beylkin *et al.* [8]. Using their examples #1 and #2 called here "BCR1" and "BCR2", test matrices of size 128×128 were multiplied in their natural domain, the wavelet domain using the standard forms of the matrices (SFM), and the wavelet domain using the non-standard forms of the matrices (NFM), both with thresholding ($t = 1 \times 10^{-4}$) and without thresholding ($t = 0$). Table I summarizes the errors computed relative to multiplication in the natural domain assumed to be the correct product. Causal peak non-aligned and anti-causal peak near-aligned phase variants are abbreviated with the labels "Null" and "Peak" respectively. Tables II and III list the data compression numbers $\mathcal{N}_f^2$ for a range of values of the fraction f (see [30] for the definition of $\mathcal{N}_f^2$) as well as the relative fractional change (RFC) in $\mathcal{N}_f^2$ for the Peak variant in comparison with the Null variant. The test matrix BCR1 in both standard and non-standard forms provides an example for which the Null and Peak convolution phase variants do not impact the compression of the matrix operator. However, the test matrix BCR2 in both standard and non-standard forms provides an example for which the different phase variants do affect the compression of the matrix operator with the Peak variant improving the compression relative to the Null variant.

IV. Discussion

A. Wavelet Transform Algorithms

As the number of applications and use of wavelet transforms continue to grow, so does the number of classes and variations of wavelet transform algorithms. All of these algorithms incorporate a convolution with a kernel in some implementation, typically, as part of an iterated filter bank. In contrast to implementations of the classical Fourier transform where mathematically there is at most a choice of sign and normalization constant in the complex exponential kernel, for wavelet transform algorithms there are multiple choices including the signs, phases, normalization constants of the wavelet kernels as well as the phase shifts of each of the filters in the wavelet filter bank. These algorithmic details, however, are usually not reported in the literature albeit with certain exceptions such as the FBI fingerprint image compression standard [6].

Nevertheless, it is necessary to specify such details in order to insure the reproducibility of results output by each algorithm regardless of its implementation by any programmer working in any language or any engineer designing any DSP chip. This report has developed a specification and evaluation methodology which includes an itemized list of choices that must be stated clearly in order to insure the reproducibility of a sequence of transform coefficients generated by a specific wavelet transform algorithm. Moreover, this report has presented a simple yet novel solution to the phase peak alignment problem for wavelet transforms (see Section II-B and Figures 1 and 2). The general principles of this solution can be applied in various specific forms to both non-subsampled and critically subsampled wavelet transforms and to both symmetric and asymmetric wavelet filters. In particular, it has been applied to the methods of Beylkin *et al.* [8] for fast-wavelet-based matrix multiplication (see Section III-D).

The latter application demonstrated at least one test matrix example for which compression was improved by the choice of a particular convolution phase variant. However, as discussed throughout the text, unambigous specifications and standards for reporting wavelet transform algorithms have been discussed here in an effort to address *reproducibility* rather than *efficiency.* In fact, developing standards for the reproducible use of computational methods in experimental science has received attention in an editorial [31] published in 1996. The author's original perspectives on reproducibility in computational science were published [32] several years earlier in 1992.

These concerns acquire growing importance for wavelet transform algorithms as their application moves into fields where data analysis rather than data compression becomes the primary task. Nowhere is this importance more dramatic than for the analysis of biomedical signals and images with results intended ultimately for clinical diagnosis and treatment. Algorithms specified to a given standard should reproducibly yield the same analytic result for the same data regardless of programming language and computing machine. If not, then different implementations in different laboratories could yield different results leading to different diagnoses and treatments for the same patient with the same clinical test data.

B. Reproducibility Standards

The scheme outlined here has been developed as a framework for specifying and evaluating wavelet transform algorithms using three heirarchical stages. For its full elaboration, it will be necessary to complete more detailed work on the methodology for each stage. Such an elaboration has already been developed further for the first stage involving the filter bank coefficients [16]. Additional work will also be pursued for the second stage and third stage involving single-level convolutions and multi-level transforms, respectively.

This methodological framework, with its heirarchical and polymorphic structures and operators, and with its empirical tests of parameters, has been designed to be comprehensive and flexible allowing for many different variations yet sufficiently complete to insure reproducibility. Note that simpler schemes do not suffice. For example, an evaluation standard cannot be limited to reporting merely the error from tests of perfect reconstruction after transforming and inverse transforming. While these tests are necessary, they are not sufficient and do not verify that the correct sequence of transform coefficients has been generated. A similar argument applies to tests of energy conservation for energy conserving transforms because transform coefficient sequences with different signs and phases may have the same energy. Thus, there is only one way to insure reproducibility of a wavelet transform algorithm and verify its correct implementation: 1) specify it completely with a sufficiently detailed combination of mathematical equations, choices of parameters, and pseudo-code templates for the algorithm, 2) verify it by comparing results with known sequences of output transform coefficients for given input signal data, and then 3) verify the inverse transform algorithm with tests for perfect reconstruction.

The elements of any such standard should specify enough information to enable the algorithm to be implemented and to yield results reproducibly in a consistent manner independently of computing platform and programming language. In this report, elements of both minimal specifications and additional characterizations were listed for three heirarchical stages of the algorithms: 1) filter bank coefficients, 2) single-level convolutions, and 3) and iterated multi-level transforms. The *minimal specifications* are required for reproducibility of results whereas the *additional characterizations*, although informative and useful, are not. Whether an element of a reproducibility standard is declared to be in the former or the latter category can be debated.

It will depend on whether the element is considered to be a constituent of the specification or the verification. For example, error tolerance limits could be established as a requirement of the specification, or errors could be reported as a characterization of the verification.

A reproducibility standard should also provide a general framework for specifying the many different types of convolutions and their phase shift variants. Different phase shift variants, such as the near-aligned solution introduced here, could affect the results of methods based on wavelet transforms, and thus should be reported. Not every convolution type requires all of the operators and parameters of the scheme reported here. If the general framework is nevertheless retained for the specification of all cases, those operators not needed for a particular convolution type can simply be set to identity matrices **I** with zero delays. Of course, a computationally efficient implementation would eliminate this redundancy. However, the general framework developed here is designed to specify and test reproducibility rather than efficiency.

C. Scientific Reproducibility versus Repetitive Executability and Input-Output Repeatability

Digital signal processing with wavelet transform algorithms can be considered a field within computational science and engineering. Algorithm specification and evaluation methodology must be developed for this field just as analysis and reporting methodologies must be improved for experimental computational science in general. This methodological work and its publication and dissemination should be pursued in a manner entirely analogous to the promotion of statistical data analysis and reporting methodology necessary for any experimental science. This position was articulated by the author in an editorial [32] published in 1992 from which the following paragraphs are quoted.

"Publishing ambiguous algorithms permits different interpretations of what is presumed mistakenly to be the same algorithm. Not publishing clear expositions of the implementations of algorithms permits alternative implementations that may be mathematically equivalent in theory on paper but numerically different in practice on the computer."

"Failure to establish clear standard definitions and names for algorithms prevents investigators from making scientifically valid comparisons of experimental results obtained from the different implementations of the different algorithms. Algorithms should be published consistent with the standards of exposition and detail as established, for example, in numerical methods and analysis."

"Methods intended for practical computation should be published consistent with the standards of any experimental science as established, for example, in physics, chemistry, biology, and medicine: that is, sufficient information should be provided so that the experiment can be easily reproduced by another independent investigator."

Some of these points deserve amplification. Proper scientific reporting for any science requires that all materials and methods including equipment, supplies, reagents, subjects, processes, procedures, *et c.,* be sufficiently well detailed that another investigator in an independent laboratory could reproduce the experiment and obtain the same results. According to this scientific standard as applied to computational science, it is therefore mandatory not only to specify algorithms in sufficient detail but also to report the version of the software implementation and hardware platform used to perform experiments. Such mandatory reporting should apply to all software whether commercial or not, just as it does in other sciences for all reagents and equipment whether commercial or not. Only by explicit reporting of hardware and software can conflicts be resolved, bugs traced, and irreproducibility stopped.

Note that this report does not present any statement that in any way implies requisite use of the WAVBOX Software Library nor any other particular software package. Thus, the specification and evaluation methodology has only been implemented and demonstrated here with WAVBOX Software but does not require it in any way. On the contrary, the methodology can be implemented in any newly written function library *in any programming language on any computing platform* of which the latter phrase is one of the requirements for reproducibility defined here. In fact, this report does *not* call for any required algorithm that investigators must use. Instead, this report provides a methodology for *evaluating and reporting* whichever wavelet transform algorithm an investigator freely chooses to use.

Careful specification and evaluation of algorithms promotes independent verification of hypotheses, results, and proposed conclusions. Skepticism, as distinct from both cynicism and unquestioning acceptance, remains one of the hallmarks of scientific investigation. Thus, scientific hypotheses remain hypotheses until they become scientific principles accepted as a consequence of independently reproducible and verifiable experimental results. Proposed engineering methods remain only proposals until the operating characteristics of their practical use are proven through independently reproducible and verifiable implementations and applications. Even proposed mathematical theorems remain only proposals until independently and rigorously verified by other means preferably based on alternate methods of proof.

In contrast to the position advocated in this report and elsewhere by the author [32], [14], Claerbout and colleagues have promoted a different meaning for "reproducible research" and "making scientific computations reproducible" with "reproducible electronic documents" [10], [33]. They describe methods to re-execute scripts and codes that regenerate electronic documents with figures. They do not describe any means by which to verify repeatability of output results for given input data other than visual inspection of figures.

Even if their methodology provided for numerical verification of the repeatability of results, any such verification would only imply the the absence of compilation and execution or runtime errors as well as the absence of randomness in the execution of the code. Thus, if the code generates an erroneous result, this error will be repeated every time the code is run. If the code is redistributed to and reexecuted by other investigators, the same error will continue to be propagated. Merely rerunning the same code does not provide the independent verification required for reproducibility of a result in experimental science. Thus, the methodology described by Claerbout *et al.* could be more correctly termed methods to insure *repetitive executability* of code that generates electronic documents, but not methods to insure *scientific reproducibility* of algorithms and experiments in computational science.

As demonstrated by the computational experiments of Hatton, errors in scientific software remain "rather worse than we would ever dare to fear" [34, pg:38]. In fact, providing experimental confirmation justifying the concerns expressed by Taswell [32], Hatton found that "the disagreement between nine different implementations of the same published mathematical algorithms written in the same language using the same input data and same disposable parameters is much worse than anticipated" and that "the disagreement among algorithms that are not as well specified is several times worse than disagreement among those that are defined formally using mathematics" [34, pg:37]. Hatton recommends that the investigator "attempt to verify [a computational result] by at least one independent software implementation." He cautions that "simply swapping software on whose calculations you depend is inherently high-risk" and that "even when independent implementations agree, there may still be problems".

Despite the significant problems with software errors in computational science and the major flaws in the meaning of "reproducibility" advocated by Claerbout *et al.* other authors have promoted his notion of "reproducibility" without questioning it, even going so far as to equate it with "scholarship". In summarizing the Claerbout notion of "reproducibility", Buckheit and

Donoho make the following claim [11, pg:59]. "An article about computational science in a scientific publication is *not* the scholarship itself, it is merely *advertising* of the scholarship. The actual scholarship is the complete software development environment and the complete set of instructions which generated the figures." (Words emphasized in quote as per [11].) This claim has since been endorsed and republished by Mallat [12, pg:17].

However, their statements must be refuted. A scientific publication should never be advertising without the substance and content necessary for an independent investigator to scientifically reproduce the work. Scientific publication should carefully document scientific scholarship which must include a complete description and discussion of literature review, motivation, hypothesis, experimental materials methods and results, and finally conclusions based on objective arguments debating the pros and cons of either various competing hypotheses, proposed algorithms, or alternative computational methods. It should not be argued that because of lack of space in journals, the investigator cannot detail methods and must relegate them to code available separately. In fact, in most experimental sciences, it is standard practice to require appropriate citation of the prior publications that provide the specifications and verifications for methods used but not detailed within the materials and methods sections of any new publication that reports experiments using the cited methods.

Nor can scholarship be equated with simply providing code or a software environment. Nevertheless, Buckheit and Donoho further imply that if an investigator does not provide freely redistributable code, he is not doing good science [11, pg:60]. " ... one can never require researchers to publish their code. But examples like the GNU project show that very bright and able people are naturally drawn to share their intellectual works with others, and so some researchers will do it. We believe that those who do will do better science than those who don't." On the contrary, requiring complete mathematical and algorithmic specification along with numerical evaluation and verification will lead to better science rather than merely redistributing code for all the reasons already elaborated in this report. Furthermore, several simple questions refute the position of Buckheit and Donoho. Does a book or journal have worse scholarship because books are not freely redistributable? Is a chemist, biologist, or physicist a worse scientist because he does not freely redistribute his equipment, supplies, and reagents?

It has not been the intent of this report to debate the meaning of science nor how to judge its quality. However, it has been the purpose of this report to clarify notions of reproducibility in computational science using wavelet transform algorithms as an example context within which to elaborate concepts. This report concludes by summarizing several of the key concepts introduced and discussed throughout the text with the following definitions.

Repetitive executability: For a particular implementation of a particular algorithm or electronic document, verification that repeated recompilation and reexecution of the code with regeneration of the output can be performed without any compiler or runtime errors.
Input-output repeatability: For a particular implementation of a particular algorithm, verification of both repetitive executability and numerically identical repeated outputs for repeated inputs.
Scientific reproducibility: For alternative implementations in any programming language on any computing machine of a particular algorithm specified unambiguously with mathematical equations and pseudo-code templates, verification of numerically equivalent repeated outputs for repeated inputs within a given numerical tolerance according to the accuracy and stability of the algorithm and the precision of the machine arithmetic. For each of the alternative implementations, both repetitive executability and input-output repeatability must hold true.

A more complete heirarchy of reproducibility in computational science and engineering must incorporate consideration of other important issues beyond specification and verification of a particular algorithm. For example, related to that algorithm, there could be analytic equivalences for mathematical identities with alternate right hand sides for the same left hand side, as well as alternate numerical discretizations for the same analytic identity if continous. These concepts will be elaborated further in future work.

References

[1] Jelena Kovacevic and Ingrid Daubechies, "Scanning the special issue on wavelets," *Proceedings of the IEEE*, vol. 84, no. 4, pp. 507–509, Apr. 1996. I

[2] Wim Sweldens, "Wavelets: What next?," *Proceedings of the IEEE*, vol. 84, no. 4, pp. 680–685, Apr. 1996. I

[3] Mark J. Shensa, "The discrete wavelet transform: Wedding the à trous and Mallat algorithms," *IEEE Transactions on Signal Processing*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992. I

[4] Olivier Rioul and Pierre Duhamel, "Fast algorithms for discrete and continuous wavelet transforms," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 569–586, Mar. 1992. I, II-B

[5] Carl Taswell and Kevin C. McGill, "Algorithm 735: Wavelet transform algorithms for finite-duration discrete-time signals," *ACM Transactions on Mathematical Software*, vol. 20, no. 3, pp. 398–412, Sept. 1994. I, II-B, II-B, II-B, II-C

[6] Jonathan N. Bradley and Christopher M. Brislawn, "The FBI wavelet/scalar quantization standard for gray-

scale fingerprint image compression," Technical Report LA-UR-93-1659, Los Alamos National Laboratory, Los Alamos, NM, 1993. I, IV-A

[7] Christopher M. Brislawn, "Classification of nonexpansive symmetric extension transforms for multirate filter banks," *Applied and Computational Harmonic Analysis*, vol. 3, no. 4, pp. 337–357, Oct. 1996. I, II-B

[8] G. Beylkin, R. Coifman, and V. Rokhlin, "Fast wavelet transforms and numerical algorithms I," *Communications on Pure and Applied Mathematics*, vol. 44, no. 2, pp. 141–183, 1991. I, III-D, IV-A

[9] Fritz Keinert, "Biorthogonal wavelet for fast matrix computations," *Applied and Computational Harmonic Analysis*, vol. 1, no. 2, pp. 147–156, Mar. 1994. I

[10] Jon Claerbout and Martin Karrenbach, "Electronic documents give reproducible research a new meaning," in *Soc. Expl. Geophys. Expanded Abstracts*, 1992, pp. 601–604, Proceedings of the 62nd Annual International Meeting. I, IV-C

[11] Jonathan B. Buckheit and David L. Donoho, "WaveLab and reproducible research," in *Wavelets and Statistics*, Anestis Antoniadis and Georges Oppenheim, Eds. 1995, vol. 103 of *Lecture Notes in Statistics*, pp. 55–81, Springer Verlag. I, IV-C, IV-C, IV-C

[12] Stephane G. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, San Diego, CA, 1998. I, IV-C

[13] Carl Taswell, "Wavelet transform convolution versions for fast wavelet based numerical algorithms," in *Abstracts of Talks at Conference on Wavelets, Relations with Operators and Applications*, Xingde Dai, Ed., July 1996, p. 27, available from xdai@uncc.edu. I

[14] Carl Taswell, "Specifications and standards for reproducibility of wavelet transforms," in *Proceedings of the International Conference on Signal Processing Applications and Technology*. Oct. 1996, pp. 1923–1927, Miller Freeman. I, IV-C

[15] Carl Taswell, *The Systematized Collection of Wavelet Filters Computable by Spectral Factorization of the Daubechies Polynomial*, Computational Toolsmiths, http://www.toolsmiths.com, 1997. II, 1, 1, III-A

[16] Carl Taswell, "Empirical tests for the evaluation of multirate filter bank parameters," *IEEE Transactions on Circuits and Systems II. Analog and Digital Signal Processing*, Submitted March 1998; available at http://www.toolsmiths.com. II, 1, III-A, IV-B

[17] Carl Taswell, *Handbook of Wavelet Transform Algorithms*, Birkhauser Boston, 1998, in preparation. II, II-B, II-E

[18] Carl Taswell, WAVBOX *Software Library and Reference Manual*, Computational Toolsmiths, http://www.wavbox.com, 1993–97. II, II-B, II-E

[19] Carl Taswell, "WAVBOX 4: A software toolbox for wavelet transforms and adaptive wavelet packet decompositions," in *Wavelets and Statistics*, Anestis Antoniadis and Georges Oppenheim, Eds. 1995, vol. 103 of *Lecture Notes in Statistics*, pp. 361–375, Springer Verlag, Proceedings of the Villard de Lans Conference November 1994. II, II-E

[20] Kambiz Nayebi, Thomas P. Barnwell III, and Mark J. T. Smith, "Time-domain filter bank analysis: A new design theory," *IEEE Transactions on Signal Processing*, vol. 40, no. 6, pp. 1412–1429, June 1992. 1, III-A

[21] Olivier Rioul, "A discrete-time multiresolution theory," *IEEE Transactions on Signal Processing*, vol. 41, no. 8, pp. 2591–2606, Aug. 1993. 1, II-B

[22] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall Signal Processing Series. P T R Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993. II-B, II-B

[23] Kevin C. McGill and Carl Taswell, “Length-preserving wavelet transform algorithms for zero-padded and linearly-extended signals,” Tech. Rep., Veterans Affairs Medical Center, Palo Alto, CA, 1992. II-B

[24] Kevin C. McGill and Carl Taswell, “Wavelet transform algorithms for finite-duration discrete-time signals,” in *Progress in Wavelet Analysis and Applications*, Yves Meyer and Sylvie Roques, Eds. 1993, pp. 221–224, Editions Frontieres, Proceedings of the International Conference on Wavelets and Applications, Toulouse France, June 1992. II-B

[25] Albert Cohen, Ingrid Daubechies, and Pierre Vial, “Wavelets on the interval and fast wavelet transforms,” *Applied and Computational Harmonic Analysis*, vol. 1, no. 1, pp. 54–81, Dec. 1993. II-B

[26] The MathWorks, Inc., Natick, MA, *MATLAB Reference Guide*, Aug. 1992. II-E

[27] Ingrid Daubechies, *Ten Lectures on Wavelets*, Number 61 in CBMS-NSF Series in Applied Mathematics. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1992. III-A

[28] Carl Taswell, “Algorithms for the generation of Daubechies orthogonal least asymmetric wavelets and the computation of their Holder regularity,” Tech. Rep., Scientific Computing and Computational Mathematics, Stanford University, Stanford, CA, Aug. 1995. III-A

[29] Carl Taswell, “Computational algorithms for Daubechies least-asymmetric, symmetric, and most-symmetric wavelets,” in *Proceedings of the International Conference on Signal Processing Applications and Technology.* Sept. 1997, pp. 1834–1838, Miller Freeman. III-A

[30] Carl Taswell, “Satisficing search algorithms for selecting near-best bases in adaptive tree-structured wavelet transforms,” *IEEE Transactions on Signal Processing*, vol. 44, no. 10, pp. 2423–2438, Oct. 1996. III-D

[31] George Cybenko and Harry Wijshoff, “Scientific standards for experimental computing,” *IEEE Computational Science & Engineering*, vol. 3, no. 3, pp. 1, Fall 1996. IV-A

[32] Carl Taswell, “Improving research productivity,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 11, no. 3, pp. 5, Sept. 1992. IV-A, IV-C, IV-C, IV-C

[33] Martin Karrenbach Matthias Schwab and Jon Claerbout, “Making scientific computations reproducible,” Tech. Rep., sepwww.stanford.edu/pub/redoc/cip.html, 1997. IV-C

[34] Les Hatton, “The T experiments: Errors in scientific software,” *IEEE Computational Science & Engineering*, vol. 4, no. 2, pp. 27–38, April-June 1997. IV-C, IV-C

TABLE I

Errors for Fast-Wavelet-Based Matrix Multiplication

Threshold	$t = 0$		$t = 1 \times 10^{-4}$	
Phase	Null	Peak	Null	Peak
SFM	1.76e-12	1.85e-12	1.73e-4	1.95e-4
NFM	2.29e-12	2.69e-12	2.07e-4	1.90e-4

TABLE II

Data Compression Numbers $\mathcal{N}_f^2$ for Standard Forms

Matrix	BCR1			BCR2		
Phase	Null	Peak	RFC	Null	Peak	RFC
$f = 0.500$	169	169	0.000	12	11	0.083
$f = 0.900$	591	584	0.012	85	66	0.224
$f = 0.950$	734	725	0.012	138	103	0.254
$f = 0.990$	1246	1239	0.006	300	263	0.123
$f = 0.999$	1975	1952	0.012	652	588	0.098

TABLE III

Data Compression Numbers $\mathcal{N}_f^2$ for Non-Standard Forms

Matrix	BCR1			BCR2		
Phase	Null	Peak	RFC	Null	Peak	RFC
$f = 0.500$	229	230	-.004	12	11	0.083
$f = 0.900$	811	808	0.004	118	87	0.263
$f = 0.950$	979	975	0.004	233	158	0.322
$f = 0.990$	1320	1316	0.003	595	386	0.351
$f = 0.999$	1916	1898	0.009	1179	931	0.210

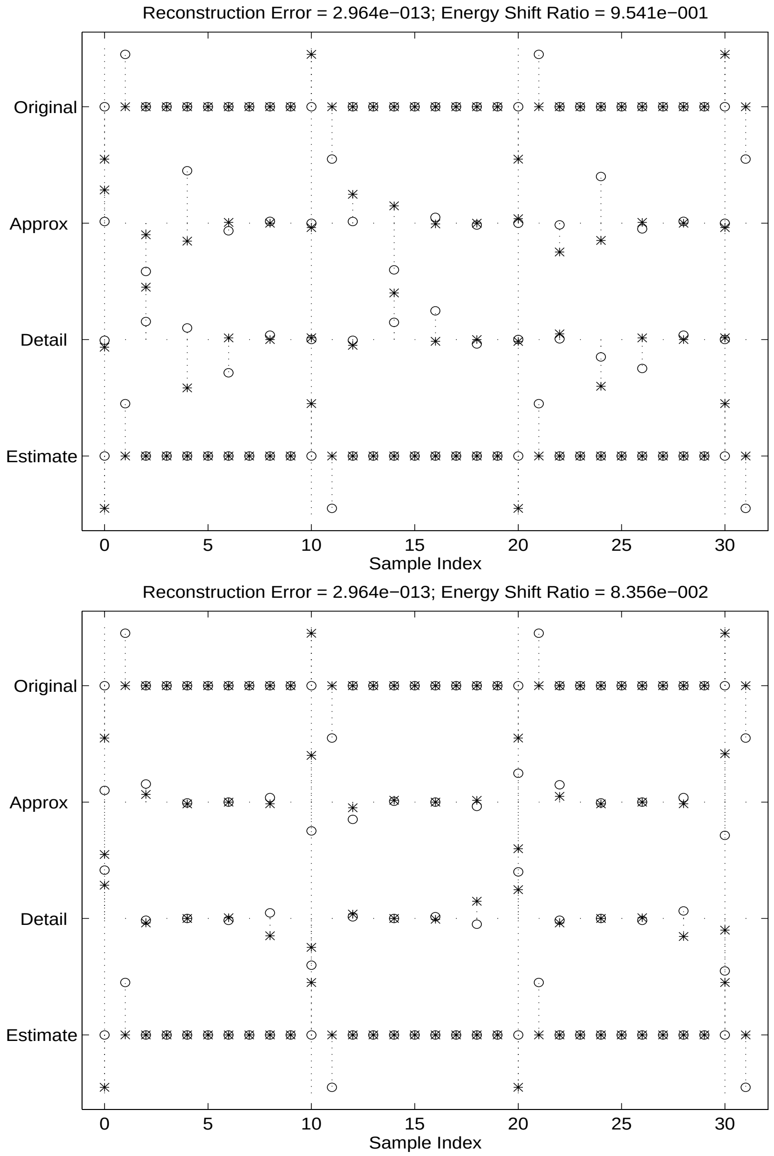

Fig. 1

Single-level decomposition and reconstruction with circularly-periodized convolution for quadruple M-spike test signal. Channel 0: “*”; Channel 1: “o”. Top: peak non-aligned phase; Bottom: peak near-aligned phase.

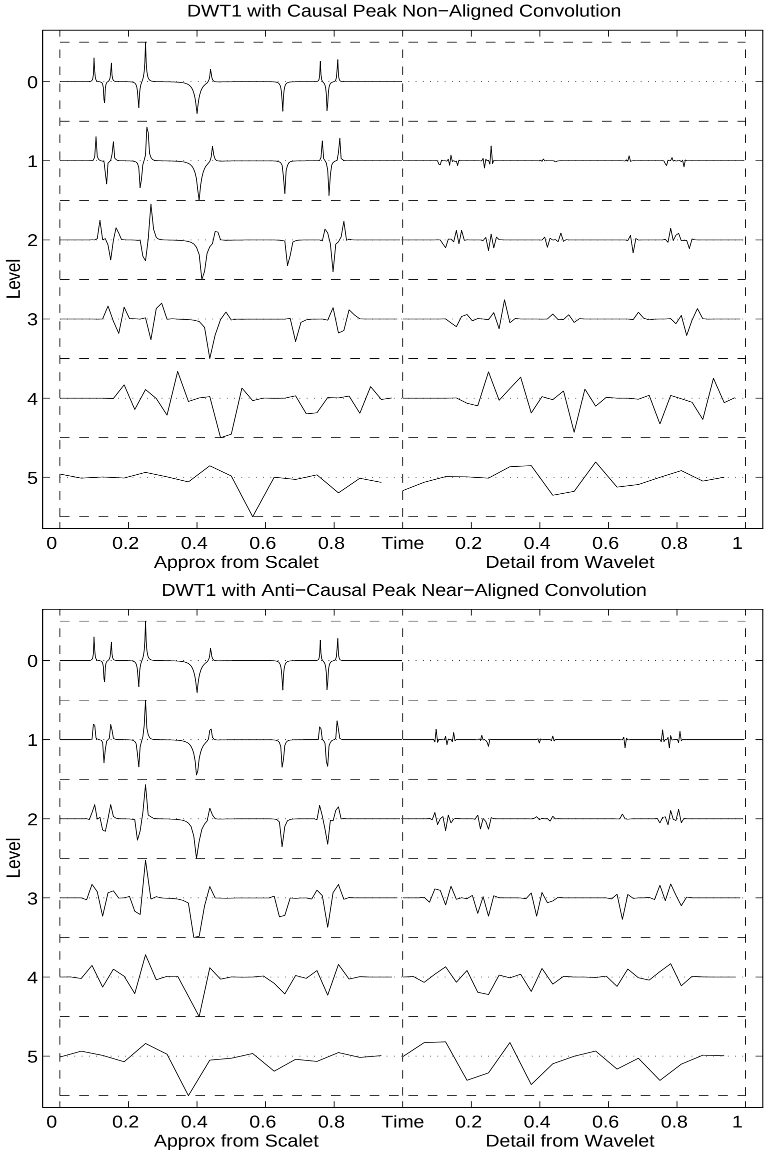

Fig. 2

Multi-level transform with circularly-periodized convolution for “peaks & chasms” test signal. Top: peak non-aligned phase; Bottom: peak near-aligned phase.

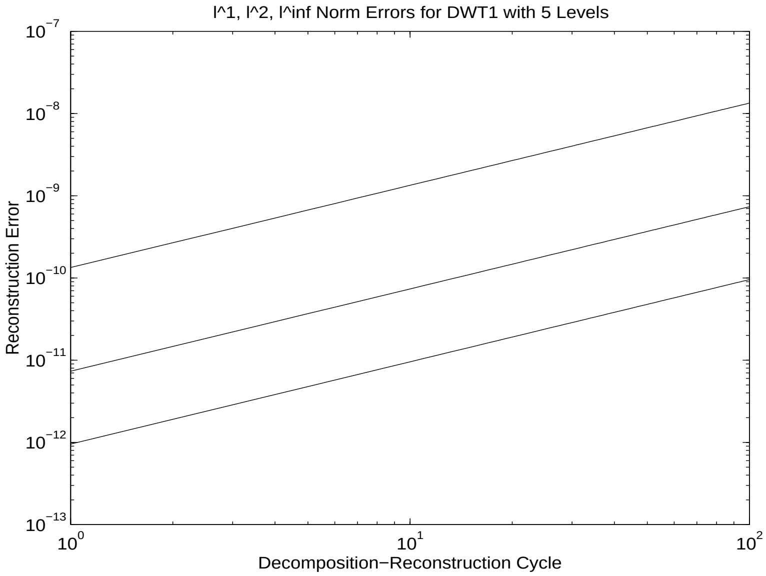

Fig. 3

PLOTS OF $\log_{10} \mathcal{E}(k)$ VERSUS $\log_{10} k$ FOR ℓ^1, ℓ^2, AND ℓ^∞ ERROR NORMS FOR A WAVELET TRANSFORM WITH $L = 5$.