Discrete Wavelet Transforms in the Large Time-Frequency Analysis Toolbox for Matlab/GNU Octave

Zdeněk Průša, Peter L. Søndergaard, Austrian Academy of Sciences
Pavel Rajmic, Brno University of Technology

The discrete wavelet transform module is a recent addition to the Large Time-Frequency Analysis Toolbox (LTFAT). It provides implementations of various generalizations of the well-known Mallat's algorithm (iterated filterbank) such that completely general filterbank trees, dual-tree complex wavelet transforms and wavelet packets can be created. The resulting transforms can be equivalently represented as filterbanks and analyzed as filterbank frames using fast algorithms.

Categories and Subject Descriptors: G.1.2 [**Approximation**]: Wavelets and fractals; H.5.5 [**Sound and Music Computing**]: Signal analysis, synthesis, and processing

General Terms: Algorithms, Documentation

Additional Key Words and Phrases: Filterbanks, finite frames, discrete wavelet transform, discrete wavelet packets, dual-tree complex wavelet transform, GNU Octave, Matlab

ACM Reference Format:
Zdeněk Průša, Peter L. Søndergaard, Pavel Rajmic, 2015. Discrete Wavelet Transform in Large Time-Frequency Analysis Toolbox for Matlab/GNU Octave *ACM Trans. Math. Softw.* 9, 4, Article 1 (March 2010), 24 pages.
DOI:http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Historically, the wavelet transform and filterbanks were connected trough the theory of Multiresolution Analysis (MRA) [Mallat 1989]. The MRA wavelet basis functions do not admit a closed form but they are created recursively using perfect reconstruction elementary filterbanks. MRA-based wavelet bases usually exhibit only a dyadic scale sampling of the underlying wavelet transform but the fact that they form bases in conjunction with existence of the fast Mallat's algorithm ensured a widespread adoption of a wavelet transform in such a form. For an introduction to the wavelet transform see e.g. [Strang and Nguyen 1997; Burrus et al. 2013].

Many scientists had focused on extending the MRA concept or the Mallat's algorithm itself trying to overcome shortcomings of the original MRA-based wavelet transform design. Such efforts were e.g. M-band wavelet transforms [Steffen et al. 1993],

---

This work was supported by the Austrian Science Fund (FWF) START-project FLAME ("Frames and Linear Operators for Acoustical Modeling and Parameter Estimation"; Y 551-N13) and by the Ministry of Education, Youth and Sport of the Czech Republic (MŠMT) National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.
Author's addresses: Zdeněk Průša, Austrian Academy of Sciences, Acoustics Research Institute, Wohllebengasse 12–14, 1040 Vienna, Austria, email: zdenek.prusa@oeaw.ac.at; Pavel Rajmic, Brno University of Technology, Faculty of Electrical Engineering and Communication, Dept. of Telecommunications, Technická 12, 616 00 Brno, Czech Republic, email: rajmic@feec.vutbr.cz; Peter L. Søndergaard, (Current address) Oticon A/S, Kongebakken 9, 2765 Smørum, Denmark, email: pesg@oticon.dk.

DOI:http://dx.doi.org/10.1145/0000000.0000000

wavelet packets [Coifman and Wickerhauser 1992], undecimated wavelet transforms [Holschneider et al. 1990], framelets [Daubechies et al. 2003] and the dual-tree wavelet transform [Kingsbury 2000] and others.

After the initial enthusiasm faded, the software support started lagging behind the theory being developed and nowadays it seems that most of the once popular toolboxes are not actively maintained, or they are even no longer available for download. Moreover, new scientific findings in the field of discrete wavelet transforms has become scattered between many small toolboxes of varying quality, usually meant only as a proof-of-concept implementation accompanying a research paper or a book.

Several existing wavelet toolboxes are worth mentioning. In the following list, the year in brackets is the year of the last official release.

— Uvi_Wave toolbox (1996), no official webpage [Prelcic et al. 1996], GPL
— Wavelab (2005) http://statweb.stanford.edu/~wavelab, custom GPL-like license
— WavBox (2012) http://www.toolsmiths.com/Wavelet/WavBox, commercial.
— Official Matlab Wavelet Toolbox™ (2015) http://www.mathworks.com/products/wavelet, commercial.

Although most of the wavelet theory lean on the theory of frames, not much attention has been paid to the frame properties of the filterbank structures that the transforms are actually computed with. Initially, the filterbanks were meant only as a tool for computing approximate coefficients of a wavelet series. More precisely, none of the toolboxes treat any of the wavelet transforms as frames.

The discrete formalism of MRA in $\ell^2(\mathbb{Z})$ was introduced in [Rioul 1993] and LTFAT follows these ideas (to some extent) and extends them while relying on finite frame theory in $\mathbb{C}^L$.

LTFAT (version 2.1.0 released on 6. 5. 2015) tries to bridge the gap in the scattered wavelet software by collecting wavelet transform generalizations, while providing a unified interface to all of them and by providing tools for analyzing their frame properties. Its aim is to provide a self-contained, modern, fast and easy to use toolbox serving as a solid base for further scientific developments.

We believe that such an effort can only be useful if the toolbox is available freely, therefore LTFAT is an open-source software licensed under a permissible GPLv3 licence and it can be obtained free of charge at http://ltfat.sourceforge.net. GNU Octave, a free Matlab alternative, is fully supported for the same reason. In fact, LTFAT has been an official Octave-Forge package since 2013.

Apart from discrete wavelets, LTFAT covers a large collection of time-frequency transforms, an object-oriented frames framework and support for real-time audio processing directly from Matlab/GNU Octave. Moreover, computationally intensive routines are programmed duplicately in the C (backend library and MEX files) and C++ (OCT files) languages to achieve speedup. Interested readers can find more details in the overview papers [Søndergaard et al. 2012; Průša et al. 2014].

The purpose of this paper is to describe the algorithms used in the wavelets module and establish the connection to finite frames in $\mathbb{C}^L$. Here we list features not to be commonly found in other toolboxes:

— Free and open-source software.
— Extensible wavelet filter database.
— No restriction to critically subsampled two-channel cases.
— Support for creating completely custom filterbank trees.
— Routines for casting filterbank trees to an identical non-iterated filterbanks.
— Calculation of frame bounds of any shape of wavelet filterbank trees.
— Online documentation including examples.

— Fully supported in GNU Octave 3.8.0 and above.
— Possibility to use discrete wavelets in the frames framework which offers a common interface for most transforms in LTFAT.

To be complete, there are still areas from the wavelet theory the toolbox is lacking:

— Boundary-adjusted filters (see e.g. WaveLab) as they can only be constructed for orthonormal wavelet filters.
— Advanced multidimensional transforms, as the toolbox is primarily devoted to one-dimensional signal processing.
— The discretized variant of the continuous wavelet transform with classical wavelets such as Mexican hat, Morlet wavelet etc. The reason is that the general approach to discretization does not ensure a perfect reconstruction and therefore the mathematical frame abstraction is no longer applicable.
— A "lifting" form of the basic wavelet filterbanks [Sweldens 1996]. The lifting scheme was introduced as an alternative to the basic wavelet filterbanks both as a design and a computational procedure. It might be included in future releases.

The rest of the paper is organized as follows. Section 1.2 establishes basic concepts of finite frame theory, sections 1.3 and 1.4 define basic properties of multirate filterbank systems, which serve for both the computation and the analysis of transform properties in the subsequent sections. Sections 2, 3, 4, 5, 6 introduce all variants of wavelet type transforms currently supported in LTFAT, while section 7 wraps up with the description of an algorithm for evaluating frame properties of all the described transforms.

1.1. Notation

Throughout this paper, we will be working exclusively with finite-length signals denoted by a bold lower-case letter, represented as a column vector $\mathbf{f} = [\mathbf{f}(0), \mathbf{f}(1), \dots \mathbf{f}(L-1)]^T$ belonging to $\mathbb{C}^L$ or $\mathbb{R}^L$ with periodic indexing such that $\mathbf{f}(l + kL) = \mathbf{f}(l)$ for $l, k \in \mathbb{Z}$. Such a treatment effectively makes the signal one period of an infinite, periodic signal, which, although it causes sometimes unnatural behaviour near the signal boundaries, makes the algorithms and the mathematical treatment clearer. Matrices will be denoted by bold capital letters e.g. $\mathbf{F}$ and a conjugate transpose will be denoted as $\mathbf{F}^*$. The identity matrix will be denoted by $\mathbf{I}$. The scalar product on $\mathbb{C}^L$ is defined as $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{l=0}^{L-1} \mathbf{x}(l)\mathbf{y}(l)^*$ and the induced norm as $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$. A circular convolution as $(\mathbf{x} \circledast \mathbf{y})(n) = \sum_{l=0}^{L-1} \mathbf{x}(l)\mathbf{y}(n-l)$ for $n \in \{0, 1, \dots, L-1\}$. Upsampling by a factor of a is a map $\uparrow_a$: $\mathbb{C}^N \to \mathbb{C}^{aN}$ such that $(\uparrow_a \mathbf{x})(an) = \mathbf{x}(n)$ for $n \in \{0, 1, \dots, N-1\}$ and zero at the other positions. Downsampling by a factor a is a map $\downarrow_a$: $\mathbb{C}^{aN} \to \mathbb{C}^N$ such that $(\downarrow_a \mathbf{x})(n) = \mathbf{x}(an)$ for $n \in \{0, 1, \dots, N-1\}$. A vector reflection and conjugation (involution) will be denoted by $\overline{\mathbf{x}}(l) = \mathbf{x}(-l)^*$ for $l \in \{0, \dots, L-1\}$ assuming the periodic indexing. The discrete Fourier transform of $\mathbf{f}$ will be defined as $\widehat{\mathbf{f}}(n) = (\mathcal{F}^*\mathbf{f})(n) = \sum_{l=0}^{L-1} \mathbf{f}(l) \exp(-\mathrm{i}2\pi nl/L)$ and the inverse transform as $\mathbf{f}(l) = (\mathcal{F}\widehat{\mathbf{f}})(l) = 1/L \sum_{n=0}^{L-1} \widehat{\mathbf{f}}(n) \exp(\mathrm{i}2\pi nl/L)$. The positive remainder of division a/b is denoted as $(a \bmod b)$ and, finally, $\operatorname{lcm}(a_0, a_1, \dots, a_{M-1})$ denotes the least common multiple of $a_0, \dots, a_{M-1}$ such that $\operatorname{lcm}(a_0, a_1, \dots, a_{M-1}) = \operatorname{lcm}(\operatorname{lcm}(a_0, a_1), a_2), \dots$.

Toolbox conventions. In the actual implementation in Matlab/Octave, the signal does not have to be a column vector. When a single row vector is used, it is transposed internally, for matrices the operation is *broadcast*, i.e. applied to each column or optionally to each row if the function supports specification of the *dim* parameter.

The LTFAT toolbox functions will be referred to in a typewriter font e.g. `funname`. The complete description of functions is available at the documentation webpage http://ltfat.sourceforge.net/doc/ or via running `help funname`.

A custom function input argument parser `ltfatarghelper`[1] was included in LTFAT to make the function calls more versatile and clear. It supports positional arguments, key-value pairs and string flag groups and it allows specifying default values for them. Such features are not commonly supported in Matlab/Octave by default.

Algorithms conventions. In the description of the algorithms we make use of dynamic array constructs such as a FIFO queue and LIFO stack. Operations defined on the queue will be *enqueue* meaning appending element as last one in the queue and *dequeue* meaning removal of the first element from the queue. Operations with a stack are *push* and *pop* performing addition and removal of the last element respectively.

1.2. Finite frames

The frame theory encompasses all linear, perfect reconstruction transforms. In the finite setting [Balazs 2008; Kovačević and Chebira 2008; Casazza and Kutyniok 2013], a frame for $\mathbb{C}^L$ is a set of Λ vectors in $\mathbb{C}^L$ which can be represented as columns of an $L \times \Lambda$, $(L \leq \Lambda)$ full-row rank matrix $\mathbf{F}$. We further restrict $\Lambda < \infty$ to exclude infinite matrices. Every frame $\mathbf{F}$ is equipped with frame bounds $0 < A \leq B < \infty \in \mathbb{R}$ with

$$A \left\|\mathbf{f}\right\|^2 \leq \left\|\mathbf{F}^*\mathbf{f}\right\|^2 \leq B \left\|\mathbf{f}\right\|^2 \tag{1}$$

for all $\mathbf{f} \in \mathbb{C}^L$. The optimal bounds are the tightest possible in (1) and correspond to squares of the min and max singular values of $\mathbf{F}$ respectively. In the following text A, B will denote the optimal frame bounds. A frame is called *overcomplete* if $\Lambda > L$.

The fundamental operators associated with frames are the *analysis* operator $\mathbf{F}^* : \mathbb{C}^L \to \mathbb{C}^\Lambda$, and the *synthesis* operator $\mathbf{F} : \mathbb{C}^\Lambda \to \mathbb{C}^L$. In the finite setting, they can both be represented as matrices being conjugate transpose of each other. An invertible *frame* operator ($L \times L$ matrix) is formed as concatenation of the analysis and the synthesis operators, $\mathbf{F}\mathbf{F}^*$. The frame bounds A, B are also equal to min and max eigenvalues of the frame operator respectively. Each frame is associated with a unique canonical *dual* frame such that $\mathbf{F}_\mathrm{d} = \left(\mathbf{F}\mathbf{F}^*\right)^{-1}\mathbf{F}$ and $\mathbf{F}_\mathrm{d}\mathbf{F}^* = \mathbf{F}\mathbf{F}_\mathrm{d}^* = \mathbf{I}$ ($\mathbf{F}_\mathrm{d}^*$ coincides with the right inverse of $\mathbf{F}$).

If a frame is not overcomplete, it is an orthonormal basis if $\mathbf{F}\mathbf{F}^* = \mathbf{I}$ and a general (Riesz) basis otherwise. In such cases, the signal representation in the frame coordinates $\mathbf{f} = \mathbf{F}\mathbf{c}$ is unique with $\mathbf{c} = \mathbf{F}^{-1}\mathbf{f}$ which simplifies to $\mathbf{c} = \mathbf{F}^*\mathbf{f}$ in the orthonormal basis case. The canonical dual frame is the only one possible dual frame. When a frame is overcomplete, the solution to $\mathbf{f} = \mathbf{F}\mathbf{c}$ is not unique. All possible solutions can be parametrized as $\mathbf{c} = \mathbf{F}_\mathrm{d}^*\mathbf{f} + (\mathbf{I} - \mathbf{F}_\mathrm{d}^*\mathbf{F})\mathrm{w}$ for an arbitrary vector $\mathrm{w} \in \mathbb{C}^\Lambda$ as the $(\mathbf{I} - \mathbf{F}_\mathrm{d}^*\mathbf{F})$ part denotes projection onto the null space of $\mathbf{F}$. The minimum energy solution is simply $\mathbf{c} = \mathbf{F}_\mathrm{d}^*\mathbf{f}$.

A frame is called tight if $A = B$ and $\mathbf{F}\mathbf{F}^* = A\mathbf{I}$ and Parseval tight if in addition $A = 1$. Any tight frame can be normalized such that it becomes Parseval tight. If the frame is tight, the calculation of the canonical dual frame reduces to $\mathbf{F}_\mathrm{d} = \frac{1}{A}\mathbf{F}$. A frame admits a *painless* expansion [Daubechies et al. 1986] if the frame operator is diagonal and therefore easily invertible. When a frame is not tight, the frame operator is seldom diagonal in the original domain but it can become diagonal, or at least structured conveniently in a different domain. Applying an unitary operator $\mathbf{U}$ to each frame element results in an *unitary isomorphic* frame [Casazza and Kutyniok 2013] $\mathbf{F}_\mathrm{ui} =$

[1] Since version 2.1.0 `ltfatarghelper` is also available as a MEX function.

$\mathbf{UF}$ for which $\mathbf{F}_{\mathrm{ui}}^*\mathbf{F}_{\mathrm{ui}} = \mathbf{F}^*\mathbf{F}$ and therefore the frame bounds of $\mathbf{F}$ and $\mathbf{F}_{\mathrm{ui}}$ are equal. In the latter sections we will use the discrete Fourier transform as $\mathbf{U}$.

In general, there is an infinite number of other dual frames $\mathbf{F}_{\mathrm{dd}}$ such that $\mathbf{F}_{\mathrm{dd}}\mathbf{F}^* = \mathbf{F}\mathbf{F}_{\mathrm{dd}}^* = \mathbf{I}$. This is an important property of overcomplete frames as the canonical dual frame might not keep the same structure (and therefore admit a fast algorithm) but some other dual frame might. On the other hand, the canonical dual frame plays an important role when the coefficients are modified and one needs to make a projection onto the range of $\mathbf{F}^*$ or null space of $\mathbf{F}$. Such operations are required in some iterative schemes e.g. the Griffin-Lim algorithm for a phase retrieval task [Griffin and Lim 1984] or a basis pursuit using the Alternating Direction Method of Multipliers [Boyd et al. 2011]. Canonical dual frames are also important in the context of frame multipliers [Balazs 2007].

There are iterative algorithms for calculating the inverse of the frame operator, which might be useful when the canonical dual frame does not to admit a fast algorithm and constructing it explicitly would not be feasible. Namely the Neumann series expansion and others (e.g. [Gröchenig 1993]) which can work without explicit matrices with or without the knowledge of the frame bounds.

Toolbox Conventions. In LTFAT, no matrix is explicitly created as it would not be feasible to do for any larger L (tens of thousands and more). Instead, fast algorithms are exploited to do the actual matrix-vector multiplications and inversions. All frames that will be considered admit a fast algorithm to be used in place of their operators.

1.3. Filterbanks

Filters $\mathbf{g} \in \mathbb{C}^L$ considered in this paper will be finite impulse response (FIR) which means that the actual filter coefficients $\mathbf{g}_{\mathrm{supp}} \in \mathbb{C}^{L_{\mathbf{g}}}$ are padded with zeros to L (assuming $L_{\mathbf{g}} \leq L$) and subsequently circularly shifted upwards by d samples, where d is a filter-specific initial "offset". We will use a filter generating function $\mathbf{g} = g(L)$ with $(g(L))\,((n-d) \bmod L) = \mathbf{g}_{\mathrm{supp}}(n)$ for $n \in \{0, 1, \ldots, L_{\mathbf{g}} - 1\}$ and producing zeros at other indices. We will refer to filters without L yet specified as g. When such a filter is used in a convolution, we assume L to be deduced from the other argument. A generating function for the upsampled version of such a filter will be defined as $((\uparrow_a g)(L))((a(n-d)) \bmod L) = \mathbf{g}_{\mathrm{supp}}(n)$ for n as before and the following holds: $(\uparrow_a g)(L) = \uparrow_a g(L/a)$. A filterbank is a collection of impulse responses g_m for $m \in \{0, 1, \ldots, M-1\}$, followed by channel dependent integer subsampling factors a_m. A filterbank is called *uniform* if $a_m = \ldots = a_0 = a$. The filterbank produces length N_m *subband* coefficients $\mathbf{c}_m \in \mathbb{C}^{N_m}$ such that $L = k \operatorname{lcm}(a_0, a_1, \ldots, a_{M-1})$ for some integer $k \in \mathbb{N}$:

$$\mathbf{c}_m\,(n) = (\downarrow_{a_m}(\mathbf{f} \circledast g_m))\,(n) = \sum_{l=0}^{L-1} \mathbf{f}\,(l)\,\mathbf{g}_m\,(a_m n - l)\,, \tag{2}$$

for $m \in \{0, 1, \ldots, M-1\}$, $N_m = L/a_m$ and $n \in \{0, 1, \ldots, N_m - 1\}$. Formula (2) corresponds to an analysis operator of a filterbank frame defined by $\mathbf{g}_m$, a_m as

$$\boldsymbol{\psi}_{m,n}(l) = \left\{\overline{\mathbf{g}_m}(l - a_m n)\right\}, \tag{3}$$

for $l \in \{0, 1, \ldots, L-1\}$ and n and m as above.

The adjoint operation (the synthesis operator) is given by

$$\widetilde{\mathbf{f}}\,(l) = \sum_{m=0}^{M-1} \left((\uparrow_{a_m}\mathbf{c}_m) \circledast \overline{g_m}\right)(l) = \sum_{m=0}^{M-1}\sum_{n=0}^{N_m-1} \mathbf{c}_m\,(n)\,\overline{\mathbf{g}_m}\,(a_m n - l)\,, \tag{4}$$

for l and N_m as above. In order to get a perfect reconstruction, one has to combine the analysis operator and the synthesis operator of dual filterbank frames.

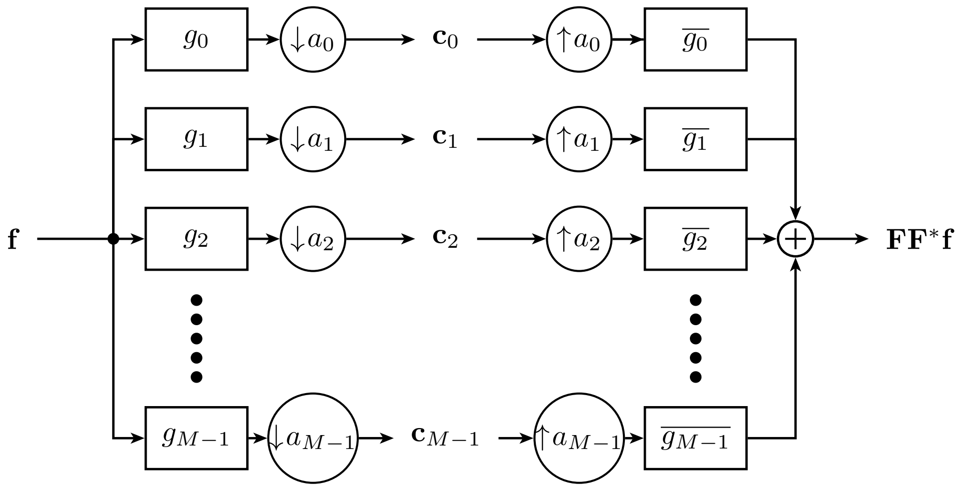

Fig. 1. Analysis and synthesis operators of a filterbank frame given as a set of filters g_m and subsampling factors a_m.

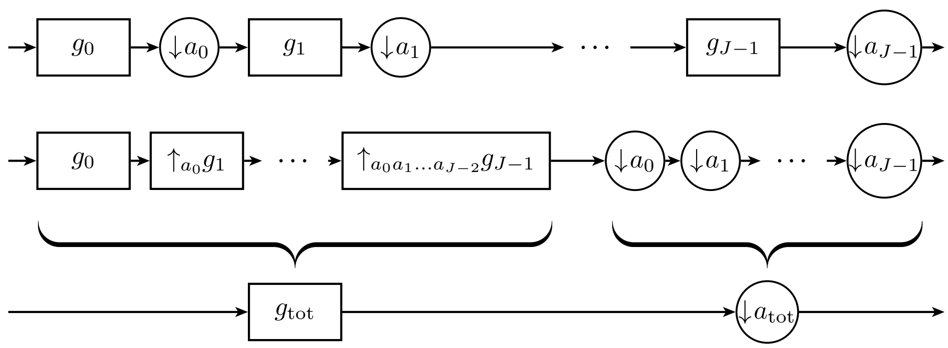

Fig. 2. Chain of filters g_k and downsampling factors a_k (top row) and the identical filter g_{tot} and downsampling factor a_{tot} (bottom row). The middle row shows an intermediate state after moving all subsampling factors to the rightmost position.

Uniform filterbanks were studied in the context of frames in [Cvetkovic and Vetterli 1998; Bölcskei et al. 2002], the results were generalized to the L-periodic sequences in [Fickus et al. 2013], which fits in the setting assumed in this paper.

Toolbox conventions. Since the set of admissible L is restricted to $L = k\,\mathrm{lcm}\,(a_0, a_1, \ldots, a_{M-1})$, $k \in \mathbb{N}$ the input signal of length L_s is padded with zeros to the proper length by default.

From a signal processing point of view, equations (2), (4) can be evaluated quickly using a time-domain polyphase implementation when working with short FIR filters or by a fast convolution using FFTs (see e.g. [Vetterli et al. 2014]). Both algorithms are well known and implemented in the computational functions called from `filterbank` and `ifilterbank`. The FIR filters are passed as a cell array of structures with fields `.h` and `.offset` representing $\mathrm{g}_{\mathrm{supp}}$ and $-d$ mentioned earlier respectively.

1.4. Multirate Identity Rules

Multirate identity rules [Vaidyanathan 1993] refer to transformations of a collection of filters and rate converters which does not change the overall transfer function.

In particular, a chain of J filters and downsamplers g_k, a_k for $k \in \{0, 1, \ldots, J-1\}$ can be transformed to a single "identical" filter g_{tot} followed by a downsampling factor a_{tot} using Algorithm 1 (see Fig. 2).

ALGORITHM 1: Create a filter identical to a chain of filters and downsamplers

Input: Chain of filters and downsamplers g_k, a_k for $k \in \{0, 1, \ldots, J-1\}$.
Output: Identical filter g_{tot} and downsampling factor a_{tot}.
Define: $a_{\mathrm{part}}(k) = \begin{cases} \prod_{n=0}^{k} a_n & \text{if } k \geq 0 \\ 1 & \text{otherwise} \end{cases}$;
Define: $L_g(k) = L_{g_0} + \sum_{n=1}^{k} (L_{g_n} - 1)\, a_{\mathrm{part}}(n-1)$;
Define: $d(k) = d_0 + \sum_{n=1}^{k} d_n a_{\mathrm{part}}(n-1)$;
$g_{\mathrm{tot}} = g_0$;
for $k \leftarrow 1$ **to** $J-1$ **do**
    $\mathbf{g}_{\mathrm{tot}} \leftarrow \left(\uparrow_{a_{\mathrm{part}}(k-1)} g_k\right) \circledast g_{\mathrm{tot}}(L_g(k))$; $g_{\mathrm{tot}} \leftarrow$ from $\mathbf{g}_{\mathrm{tot}}$ and $d(k)$;
end
$d_{\mathrm{tot}} \leftarrow d(J-1)$; $a_{\mathrm{tot}} \leftarrow a_{part}(J-1)$;

2. THE DISCRETE WAVELET TRANSFORM

The Discrete Wavelet Transform (DWT) based on MRA, considered in this paper, is most often associated with the Mallat's algorithm [Mallat 1989] and it is then termed the Fast Wavelet Transform (FWT). It consists of an iterative application of a critically subsampled basic two-channel wavelet filterbank. The algorithm was described and analyzed by many authors (see e.g. [Shensa 1992]). The most appealing property of the algorithm is its linear complexity. The main property of the transform is a one filter per octave frequency resolution and a logarithmic overall frequency scale coming from the dyadic sampling of translation and scale of the underlying continuous transform.

The M-band wavelet bases [Steffen et al. 1993] generalize DWT such that they exhibit $M-1$ linear bands per octave as M filters are used in each of the iterations. Further generalizations came from the idea of using an overcomplete filterbank frame in place of the basic wavelet filterbank (see references below, in the list of supported wavelet filters).

All the classes mentioned share the same overall iterated filterbank structure and therefore they are treated as a single type of transform. In general, a basic wavelet filterbank consist of M filters g_m, a_m with g_0 denoting the low-pass filter.

The fast algorithms for calculating the output of analysis and synthesis operators are given by Alg. 2 and Alg. 3 respectively. Alg. 2 for $J = 3$ iterations is depicted in Fig. 3. In order to get a perfect reconstruction, dual basic wavelet filters must be used in one of the operators.

The algorithm effectively follows a filterbank tree, which can be equivalently represented by a non-uniform filterbank using Alg. 1 such that each output is associated with a chain of filters and downsamplers in a path from the root node to the output. Although the identical filterbank admits a less effective algorithm in general, it allows investigating properties of the resulting subbands $\mathbf{c}_k$ like frequency bands and group delay and it also determines the effective frame vectors as in (3). In the usual cases when the frequency responses of basic filterbank filters cover distinct frequency bands ordered by increasing center frequency, the identical filterbank frequency response is ordered in the same way.

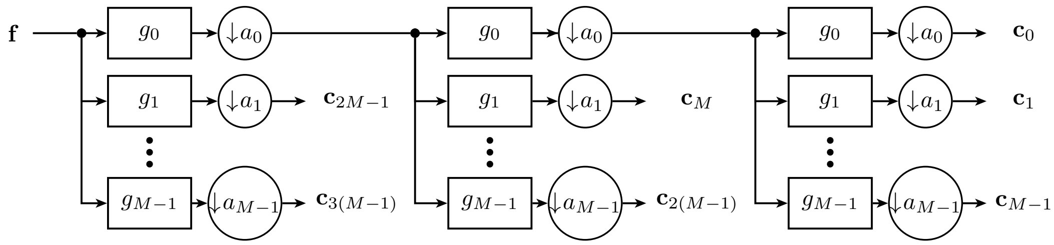

Fig. 3. Iterated filterbank structure with $J = 3$.

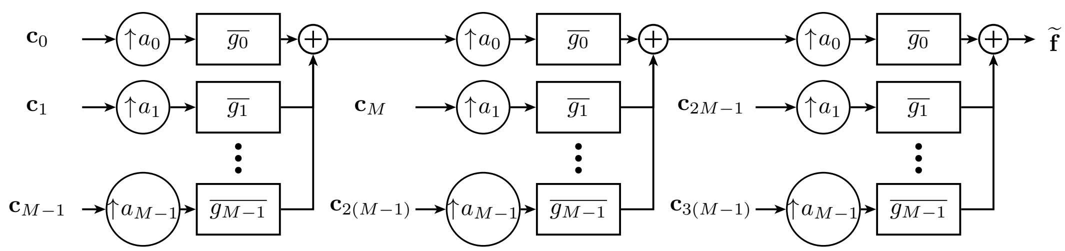

Fig. 4. Adjoint filterbank structure with $J = 3$.

ALGORITHM 2: Fast wavelet analysis

Input: Input signal $\mathbf{f} \in \mathbb{C}^{L_s}$, basic wavelet filters g_m, a_m for $m \in \{0, 1, \ldots, M-1\}$, number of levels J.
Output: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \ldots, J(M-1)\}$.

$\mathbf{a} \leftarrow \mathbf{f}$ and pad with zeros to L, the next integer multiple of a_0^J;
$k \leftarrow J(M-1)$;
for $j \leftarrow 0$ **to** $J-1$ **do**
  for $m \leftarrow M-1$ **to** 1 **do**
    $\mathbf{c}_k \leftarrow \downarrow_{a_m}(\mathbf{a} \circledast g_m)$;
    $k \leftarrow k-1$;
  end
  $\mathbf{a} \leftarrow \downarrow_{a_0}(\mathbf{a} \circledast g_0)$;
end
$\mathbf{c}_0 \leftarrow \mathbf{a}$;

Toolbox conventions. The routines implementing Algs. 2 and 3 are called `fwt` and `ifwt` respectively. Wavelet filters are stored in a direct filterbank form as defined in sec. 1.3. In fact, a pair of (dual) basic wavelet filterbanks is always stored (or computed) and the filterbanks differ only in a non-tight frame case. The basic wavelet filter impulse responses are stored in, or generated by functions with `wfilt_` prefix. LTFAT currently includes the classical ones i.e. Daubechies filters `db`, symlets `sym`, coiflets `coif`, biothogonal spline wavelet filters `spline` [Daubechies 1992] and some less common filters e.g. near-orthogonal symmetric and symmetric near-orthogonal `symorth` [Abdelnour and Selesnick 2004]. Additional supported wavelet filters are: M-band wavelets `cmband`, `mband`[2], `algmband` [Gopinath and Burrus 1995; Alkin and Caglar 1995; Lin et al.

[2]This is technically not a proper basic wavelet filterbank as the low-pass filter does not fulfill the regularity condition, but nevertheless, for a finite number of transform levels, it creates an orthonormal basis.

ALGORITHM 3: Fast wavelet synthesis

Input: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \dots, J(M-1)\}$, basic wavelet filters g_m, a_m for $m \in \{0, 1, \dots, M-1\}$, number of levels J, original input signal length L_s.
Output: Reconstructed signal $\widetilde{\mathbf{f}} \in \mathbb{C}^{L_s}$.

$\mathbf{a} \leftarrow \mathbf{c}_0$;
$k \leftarrow 1$;
for $j \leftarrow J-1$ **to** 0 **do**
    $\mathbf{a} \leftarrow (\uparrow_{a_0}\mathbf{a}) \circledast \overline{g_0}$;
    for $m \leftarrow 1$ **to** $M-1$ **do**
        $\mathbf{a} \leftarrow \mathbf{a} + (\uparrow_{a_m}\mathbf{c}_k) \circledast \overline{g_m}$;
        $k \leftarrow k+1$;
    end
end
$\widetilde{\mathbf{f}} \leftarrow \mathbf{a}(0, 1, \dots, L_s - 1)$;

2006], wavelet filters for constructing overcomplete tight or general filterbank frames `dden`, `symdden`, `symtight`, `hden`, `dgrid`, `symds`, [Selesnick 2001; Selesnick and Abdelnour 2004; Abdelnour and Selesnick 2005; Selesnick 2006; Abdelnour 2007; 2012].

The initial position (d from sec. 1.3) of the basic filters is usually not addressed in the literature since only the relative alignment of the filters makes a difference from the point of view of the MRA theory. However it might become a source of confusion in the finite setting with periodic boundary conditions as the frame vectors associated with different levels of DWT can be supported at distinct time positions causing misalignment of the coefficient subbands. Most of the basic wavelet filters are not symmetric and therefore do not admit a distinctive "center" point. Some toolboxes simply do not treat this detail, others employ ad-hoc methods for finding the center point while keeping the relative filter alignment intact but the methods might be suboptimal in some cases. In LTFAT, the default position of the basic filters was hand-tunned to minimize such ambiguity.

3. GENERAL FILTERBANK TREES AND WAVELET PACKETS

By a filterbank tree we understand a tree-shaped, oriented (no backwards connections) interconnection of basic wavelet filterbanks. DWT can be regarded as a special shape of a filterbank tree where only the low-pass output is decomposed up to depth J and all the nodes contain the same basic filterbank. In general filterbank trees, any further branching is possible and even the filterbanks in the nodes might differ; see Fig. 5 for a filterbank tree example. This results in a modular division of the frequency band. Alg. 4 acts as the analysis operator of a filterbank tree frame, whereas Alg. 5 acts as the synthesis operator. In order to obtain a perfect reconstruction, the corresponding nodes in the analysis tree and the synthesis tree have to be dual filterbanks.

Wavelet packets [Coifman and Wickerhauser 1992] share the same tree shape with the general filterbanks, but, in addition, all the intermediate outputs of all nodes are kept as additional subbands. In the literature, there is usually no distinction between wavelet filterbank trees and wavelet packets, but we have made the distinction to emphasize the slightly different algorithms and to be able to investigate different frame properties of both representations.

Since the highly redundant wavelet packet representation uses the same filters as the wavelet filterbank trees, it is necessary to introduce scaling of all intermediate subbands (or filterbanks which produce them) in the analysis and/or synthesis operation, in order to achieve perfect reconstruction. Three types of scaling are supported: no scaling, by $1/2$ or by $1/\sqrt{2}$. Wavelet packet analysis is performed according to Alg. 6

and the synthesis according to Alg. 7. Again, in order to obtain a perfect reconstruction, dual filterbanks have to be used in the corresponding nodes during analysis and synthesis in addition to the compatible scaling of the intermediate subbands.

ALGORITHM 4: Filterbank tree analysis

Input: Input signal $\mathbf{f} \in \mathbb{C}^{L_s}$, filterbank tree $\mathcal{T}$ with N nodes in the breadth-first order where the n-th node consists of filterbank g_m^n, a_m^n for $m \in \{0, 1, \dots, M_n - 1\}$. Total number of unconnected outputs of all nodes is K.
Output: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \dots, K-1\}$ in the natural ordering.

```
Determine the total subsampling factors for all outputs a_tot,k according to Alg. 1;
a ← f and pad with zeros to L, the next integer multiple of lcm(a_tot,0, a_tot,1, ..., a_tot,K−1);
kIdMap ← Alg. 20 with T;
k ← 0;
aFifo ← {} ; % an empty FIFO queue of vectors
enqueue a to aFifo;
for n ← 0 to N − 1 do
    a ← dequeue from aFifo;
    for m ← 0 to M_n − 1 do
        b ←↓_{a_m^n}(a ⊛ g_m^n);
        if node n has a next node connected at index m then
            enqueue b to aFifo;
        else
            c_kIdMap(k) ← b;
            k ← k + 1;
        end
    end
end
```

As in the case of DWT in sec. 2, the filterbank tree can be transformed to an identical filterbank using Alg. 1 for each of the paths from the root to k-th output of the tree. The resulting frequency bands are in the *natural* order (see sec. 3.1).

The tree can be build completely custom using functions `wfbtinit`, `wfbtput` and `wfbtremove` (see examples in the documentation of the respective functions). There are two predefined tree shapes: DWT tree (`'dwt'` flag) and full tree (`'full'` flag) decomposition up to a specified depth J repeating the same basic filterbank in each node. Moreover, using a trick due to [Bayram and Selesnick 2008] allows creating a 2^k-band wavelet transforms from a basic two channel elementary filterbanks by using k level full decomposition trees as levels of a 2^k-band DWT filterbank tree. The symbol J then denotes the number of levels of the 2^k-band transform. Such a filterbank tree can be created by using `'doubleband'` for $k = 2$, `'quadband'` ($k = 3$) and `'octaband'` ($k = 4$) flags in the computational routines.

A traditional way of creating a particular filterbank tree adapted to an input signal is based on evaluating a cost function of wavelet packet subbands. The procedure is referred to the best-basis selection [Wickerhauser 1994], which was generalized to non-additive cost functions in [Taswell 1994]. Both cases are addressed in Alg. 8. It is restricted to orthonormal basic filterbanks. Any filterbank subtree chosen by the algorithm forms an orthonormal basis.

ALGORITHM 5: Filterbank tree synthesis

Input: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \ldots, K-1\}$, filterbank tree $\mathcal{T}$ with N nodes in the breadth-first order where the n-th node consists of filterbank g_m^n, a_m^n for $m \in \{0, 1, \ldots, M_n - 1\}$ and the number of all unconnected outputs of all nodes is K, original input signal length L_s.
Output: Reconstructed signal $\widetilde{\mathbf{f}} \in \mathbb{C}^{L_s}$.
$kIdMap \leftarrow$ Alg. 20 with $\mathcal{T}$;
$k \leftarrow K - 1$;
$aFifo \leftarrow \{\}$; `% an empty FIFO queue of vectors`
for $n \leftarrow N-1$ **to** 0 **do**
  $\mathbf{b} \leftarrow 0$;
  for $m \leftarrow M_n - 1$ **to** 0 **do**
    if *node* n *has a next node connected at index* m **then**
      $\mathbf{a} \leftarrow$ dequeue from $aFifo$;
    else
      $\mathbf{a} \leftarrow \mathbf{c}_{kIdMap(k)}$;
      $k \leftarrow k - 1$;
    end
    $\mathbf{b} \leftarrow \mathbf{b} + \uparrow_{a_m^n}(\mathbf{a} \circledast \overline{g_m^n})$;
  end
  enqueue $\mathbf{b}$ to $aFifo$;
end
$\widetilde{\mathbf{f}} \leftarrow \mathbf{b}(0, 1, \ldots, L_s - 1)$;

ALGORITHM 6: Wavelet packet analysis

Input: Input signal $\mathbf{f} \in \mathbb{C}^{L_s}$, filterbank tree $\mathcal{T}$ with N nodes in the breadth-first order, where the n-th node consist of a filterbank g_m^n, a_m^n for $m \in \{0, 1, \ldots, M_n - 1\}$, scaling of intermediate subbands $s \in \{1, 1/\sqrt{2}, 1/2\}$.
Output: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \ldots, -1 + \sum M_n\}$, stack of indexes of nodes $inIdStack$.
Determine the total subsampling factors for all outputs $a_{\mathrm{tot},k}$ according to Alg. 1;
$\mathbf{a} \leftarrow \mathbf{f}$ and pad with zeros to L, the next integer multiple of $\mathrm{lcm}(a_{\mathrm{tot},0}, a_{\mathrm{tot},1}, \ldots, a_{\mathrm{tot},K-1})$;
$k = 0$;
$inIdFifo = \{\}$; `% an empty FIFO queue of indices`
$inIdStack = \{\}$; `% an empty LIFO stack of indices`
for $n \leftarrow 0$ **to** $N-1$ **do**
  for $m \leftarrow 0$ **to** $M_n - 1$ **do**
    $\mathbf{c}_k \leftarrow \downarrow_{a_m^n}(\mathbf{a} \circledast g_m^n)$;
    if *node* n *has a next node connected at* m **then**
      $\mathbf{c}_k \leftarrow s\mathbf{c}_k$;
      enqueue k to $inIdFifo$;
      push k to $inIdStack$;
    end
    $k = k + 1$;
  end
  if $inIdFifo$ *is not empty* **then**
    $k_{\mathrm{next}} \leftarrow$ dequeue from $inIdFifo$;
    $\mathbf{a} = \mathbf{c}_{k_{\mathrm{next}}}$;
  end
end

ALGORITHM 7: Wavelet packet synthesis

Input: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \ldots, -1 + \sum M_n\}$, filterbank tree $\mathcal{T}$ with N nodes in a breadth-first order, where n-th node consists of filterbank g_m^n, a_m^n for $m \in \{0, 1, \ldots, M_n - 1\}$, stack of indexes $inIdStack$, scaling of intermediate subbands $s \in \{1/2, 1/\sqrt{2}, 1\}$, original input length L_s.
Output: Reconstructed signal $\widetilde{\mathbf{f}} \in \mathbb{C}^{L_\mathrm{s}}$.
$k = -1 + \sum M_n$;
for $n \leftarrow N - 1$ **to** 0 **do**
  if $n \neq 0$ **then** `% not a root node`
    id pop from $inIdStack$;
    $\mathbf{a} \leftarrow \mathbf{c}_{id}$;
  else
    $\mathbf{a} \leftarrow 0$;
  end
  for $m \leftarrow M_n - 1$ **to** 0 **do**
    $\mathbf{a} \leftarrow \mathbf{a} + \left(\uparrow_{a_m^n} \mathbf{c}_k\right) \circledast \overline{g_m^n}$;
    $k \leftarrow k - 1$;
  end
  if $n \neq 0$ **then** $\mathbf{c}_{id} \leftarrow s\mathbf{a}$;
end
$\widetilde{\mathbf{f}} \leftarrow \mathbf{a}(0, 1, \ldots, L_\mathrm{s} - 1)$;

ALGORITHM 8: (Near) Best basis selection

Input: $\mathbf{f} \in \mathbb{C}^{L_\mathrm{s}}$, filterbank tree $\mathcal{T}_\mathrm{in}$ built using orthonormal basic filterbanks, cost measure function $\mathcal{C}$.
Output: Pruned filterbank tree $\mathcal{T}_\mathrm{out}$, (near) best basis coefficients $\mathbf{c}$.
$\mathbf{f}_\mathrm{norm} \leftarrow \mathbf{f} / \|\mathbf{f}\|$;
$nToRemove = \{\}$; `% Empty set of indices`
$\mathbf{c}_\mathrm{in}, inIdStack \leftarrow$ wavelet packet analysis of $\mathbf{f}_\mathrm{norm}$ using Alg. 6 with $\mathcal{T}_\mathrm{in}$, scaling $s = 1$.
foreach *subband in* $\mathbf{c}_\mathrm{in}$ **do** $C_k \leftarrow \mathcal{C}(\mathbf{c}_{\mathrm{in},k})$;
$k = -1 + \sum M_n$;
if $\mathcal{C}$ *is additive* **then**
  for $n \leftarrow N - 1$ **to** 1 **do**
    `% Parent subband index`
    id pop from $inIdStack$;
    `% Combined cost of children subbands`
    $C = \sum_{m=k-M_n+1}^{k} C_m$;
    $k = k - M_n$;
    if $C_{id} \leq C$ **then**
      add n to $nToRemove$;
    else
      $C_{id} \leftarrow C$;
    end
  end
end
if $\mathcal{C}$ *is not additive* **then**
  `% Empty sets of indices`
  for $n \leftarrow N - 1$ **to** 1 **do** $chid_n \leftarrow \{\}$;
  for $n \leftarrow N - 1$ **to** 1 **do**
    `% Parent subband index`
    $id \leftarrow$ pop from $inIdStack$;
    $chid \leftarrow \{k - M_n + 1, \ldots, k\}$;
    `% Combined cost of children subbands`
    $C = \mathcal{C}(\mathbf{c}_{chid_n \cup chid})$;
    $k = k - M_n$;
    if $C_{id} \leq C$ **then**
      add n to $nToRemove$;
    else
      $chid_{id} \leftarrow chid_n \cup chid$;
    end
  end
end
Delete $nToRemove$ nodes from $\mathcal{T}_\mathrm{in}$ to obtain $\mathcal{T}_\mathrm{out}$;
$\mathbf{c} \leftarrow$ filterbank tree analysis of $\mathbf{f}$ using. Alg. 4 with $\mathcal{T}_\mathrm{out}$;

Toolbox conventions. The filterbank trees are stored in an abstract data structure behaving like a tree. It consists of an array of nodes containing basic filterbanks from sec. 2 and two arrays defining indexes of parent and children nodes [3].

The functions implementing Alg. 4 and Alg. 5 are `wfbt` and `iwfbt` respectively. Alg. 6 and Alg. 7 are implemented in `wpfbt` and `iwpfbt` respectively. Function `wpbest` implements the best basis selection (algorithm 8).

3.1. Natural (Paley) and Frequency ordering of subbands

A filterbank tree subbands, as obtained from Alg. 4 or 6, are not ordered according to the frequency. For a full decomposition tree and nodes containing two half-band filters, this ordering is called *natural* or *Paley*. In such a case, the subbands can be shuffled to a proper frequency ordering by rewriting the subband indexes in Gray binary code. A general way of handling the frequency ordering of subbands valid for any decomposition tree shape involves reversing the order of filters in some nodes as it is described in Alg. 9. Applying the algorithm for the second time changes the frequency subband ordering back to natural. The algorithm comes from [Wickerhauser 1994], originally designed to work with filterbanks with two filters only and it was modified to correctly treat basic wavelet filterbanks with odd number of filters.

Although the filterbank tree can be build completely general, some node combinations do not result in well localized frequency bands and changing the subband ordering is not relevant in such cases.

ALGORITHM 9: Create a filterbank tree with frequency ordering of subbands

Input: Filterbank tree $\mathcal{T}_{\text{in}}$ producing subbands in the natural order. N nodes of the tree are traversed in a breadth-first order and n-th node consists of filterbank g_m^n, a_m^n for $m \in \{0, 1, \ldots, M_n - 1\}$.
Output: Filterbank tree $\mathcal{T}_{\text{out}}$ producing subbands in the frequency order.
$\mathcal{T}_{\text{out}} \leftarrow \mathcal{T}_{\text{in}}$; `% The following operations are performed on` $\mathcal{T}_{\text{out}}$.
Mark all nodes n as not reordered;
for $n \leftarrow 0$ **to** $N-1$ **do**
    if *node n is reordered and M_n is odd* **then**
        $parity \leftarrow$ even;
    else
        $parity \leftarrow$ odd;
    end
    for $m \leftarrow 0$ **to** $M_n - 1$ **do**
        if *node n has a next node k connected at m and m is $parity$* **then**
            for $p \leftarrow 0$ **to** $M_k - 1$ **do** `% Flip order of filters in node` k
                $g_p^k \leftarrow g_{M_k-1-p}^k$; $a_p^k \leftarrow a_{M_k-1-p}^k$;
            end
            Mark node n as reorderd;
        end
    end
end

[3]The data structure was not implemented as a Matlab class since Octave in its current version 3.8.2 does not fully support `classdef` yet.

3.2. Alternative Boundary Handling

Assuming the periodicity of signals is unnatural in a sense while it might result in “false” significant wavelet coefficients due to the possible discontinuities introduced by the periodic wraparound. A general approach to handling a general boundary extension is to allow for some additional coefficients which hold information about the boundaries. Such general case was analyzed in detail in [Rajmic and Průša 2014]. The main idea is to replace the circular convolution with a linear one assuming the signal being extended according to the chosen extension type. This approach however breaks the finite frame abstraction. Other boundary handling tricks can be found in [Taswell and McGill 1994] and elsewhere, but such are not yet included in the toolbox.

Toolbox conventions. The computation functions introduced so far accept additional `'zero'`, `'odd'` and `'even'` flags for zero, even or odd symmetric extensions.

4. UNDECIMATED WAVELET FRAMES, NORMALIZATION, CANONICAL DUALS

An undecimated overcomplete wavelet frame is obtained by removing the downsampling from a filterbank tree. This makes the representation translation-invariant, but on the other hand highly redundant. Transform of such a type is known under several names: stationary, maximum overlap, redundant and even continuous wavelet transform.

The undecimated transform can be achieved from a filterbank by removing the downsampling operations and upsampling the filters according to the mutirate identity property (see sec. 1.4 and in particular the middle row in Fig. 2). This filterbank shape together with a quick evaluation of the convolutions is called the *À–trous* algorithm [Holschneider et al. 1990].

Since the undecimated transforms use the same filters as their decimated counterparts, scaling in the analysis and/or synthesis step needs to be employed in order to achieve a perfect reconstruction. Three types of scaling are supported: *noscale*, *scale* and *sqrt* causing scaling of each involved filter by 1, by one over the associated subsampling factors or square roots of the previous, respectively. In order to get a perfect reconstruction, *sqrt* must be used in both analysis and synthesis or *noscale* must be used for analysis and *scale* for synthesis, or vice versa.

Algorithms 10 and 11 are valid only for a DWT filterbank tree, but an extension to general filterbanks and to wavelet packets is straightforward and they are also included in LTFAT. The *À–trous* trick exploits the fact that the upsampled filter in the convolution has only $L_{\mathbf{g}}$ nonzero samples.

ALGORITHM 10: À-trous algorithm analysis

Input: Input signal $\mathbf{f} \in \mathbb{C}^L$, basic wavelet filters g_m, a_m for $m \in \{0, 1, \dots, M-1\}$, number of levels J, scaling factors $s_m \in \{1, 1/\sqrt{a_m}, 1/a_m\}$.
Output: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \dots, J(M-1)\}$.
Scale all filters such that $g_m \leftarrow s_m g_m$;
Continue as in Alg. 2 with lines 5 and 8 changed as follows:
line 5: $\mathbf{c}_k \leftarrow \mathbf{a} \circledast \left(\uparrow_{a_0^j} g_m\right)$;
line 8: $\mathbf{a} \leftarrow \mathbf{a} \circledast \left(\uparrow_{a_0^j} g_0\right)$;

Any undecimated wavelet filterbank tree or packet can be equally represented as a simple undecimated filterbank which, in turn, can be regarded as a painless, frequency domain version of the nonstationary Gabor transform [Balazs et al. 2011].

ALGORITHM 11: À-trous algorithm synthesis

Input: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \ldots, J(M-1)\}$, basic wavelet filters g_m, a_m for $k \in \{0, 1, \ldots, M-1\}$, number of levels J, scaling factors $s_m \in \{1/a_m, 1/\sqrt{a_m}, 1\}$.
Output: Reconstructed signal $\mathbf{f} \in \mathbb{C}^L$.
Scale all filters such that $g_m \leftarrow s_m g_m$;
Continue as in Alg. 3 with lines 4 and 6 changed as follows;
line 4: $\mathbf{a} \leftarrow \mathbf{a} \circledast \overline{\left(\uparrow_{a_0^j} g_0\right)}$;
line 6: $\mathbf{a} \leftarrow \mathbf{a} + \mathbf{c}_k \circledast \overline{(\uparrow_{a_0^j} g_m)}$;

Consequently, the canonical dual frame can be obtained easily exploiting the fact that the frame operator is diagonal in the frequency domain.

The downside of this approach is that the canonical dual frame will not keep the iterated filterbank structure, but it becomes a uniform filterbank which admits a potentially less effective algorithm. In general, Alg. 12 produces the canonical dual frame valid for a single length L since the support of the dual filters might not be compact any more. A reasonable FIR approximation can usually be obtained trough truncation.

ALGORITHM 12: Find the canonical dual frame of an undecimated wavelet frame

Input: General undecimated filterbank tree $\mathcal{T}_{\text{in}}$ with K outputs, system length L, frequency ordering boolean flag ($freqOrdering$).
Output: Dual frame undecimated filterbank $\mathbf{h}_k$, $a = 1$ for $k \in \{0, 1, \ldots, K-1\}$ for a system length L.
if *freqOrdering* **then** $\mathcal{T}_{\text{in}} \leftarrow$ Alg. 9 with $\mathcal{T}_{\text{in}}$;
foreach *k-th output of* $\mathcal{T}_{\text{in}}$ **do**
  $g_k \leftarrow$ identical filter using Alg. 1;
  $\widehat{\mathbf{g}_k} \leftarrow \mathcal{F}^*(g_k(L))$;
end
for $k \leftarrow 0$ **to** $K-1$ **do**
  $\mathbf{h}_k \leftarrow \mathcal{F}\left(\widehat{\mathbf{g}_k} / \sum_{m=0}^{M-1} |\widehat{\mathbf{g}_k}|^2\right)$; `% Squaring and division are performed element-wise.`
end

Toolbox conventions:. The functions performing the undecimated versions of transforms from the previous sections are: `ufwt`, `iufwt`, `uwfbt`, `iuwfbt`, `uwpfbt` and `iuwpfbt`. The canonical dual frame can be computed using combination of functions `wfbt2filterbank` or `wpfbt2filterbank` and `filterbankdual`.

5. DUAL-TREE COMPLEX WAVELET TRANSFORM

The dual-tree complex wavelet transform (DT-$\mathbb{C}$WT) is a two-times redundant representation based on a pair of DWT filterbank trees behaving as a Hilbert transform pair. In the original design [Kingsbury 2000; 2001; 2003; Selesnick et al. 2005], all the filters are FIR and the Hilbert transform relationship is only approximate since the exact relationship would enforce an infinitely supported filters in one of the trees or a pre-processing with an infinitely supported Hilbert transformer. If the coefficients are properly combined, the overall transform is almost analytic, meaning that the effective frame vectors are complex and their Fourier transform is supported largely in the positive or negative frequency domains.

DT-$\mathbb{C}$WT has the advantage over the ordinary DWT that the subbands are almost aliasing-free, which makes the representation near translation-invariant and more robust to undesired effects during the coefficient processing. Moreover, the real input signals require only half of the coefficients to be computed since the other half is just a complex conjugate, much like the DFT of a real signal.

The original design procedure suggests to use one sample delay difference in the very first level of trees which results in a better analytic behaviour of the first levels of the transform. In fact, any basic wavelet filterbank can be used in the first level of both trees, as long as it has the same number of filters and equal downsampling rates as the rest of the tree and provided that the filters are shifted by one sample in one of the trees. In effect, such trees have to be treated as general filterbank trees as described in sec. 3.

A natural extension was proposed in [Selesnick 2004], where the dual-density and dual-tree wavelet transforms were combined. In such case, the tree nodes are three-channel filterbanks and the algorithms still apply.

The coefficients of a dual tree complex wavelet transform can be obtained from a signal by Alg. 13 and the synthesis operation is performed by Alg. 14.

As in the case of general filterbank trees, the overall dual-tree filterbank can be equally represented by a non-iterated filterbank obtained by combining multirate identity representations of both trees.

ALGORITHM 13: Dual-Tree complex wavelet analysis

Input: Input signal $\mathbf{f} \in \mathbb{C}^{L_s}$, Hilbert transform pair of basic filterbanks $g_{\mathrm{a},m}$ and $g_{\mathrm{b},m}$, a_m, first level filterbank g_m, $m \in \{0, 1, \ldots, M-1\}$, number of levels J.
Output: Complex coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \ldots, 2K-1\}$ where $K = J(M-1)+1$.

Form a DWT filterbank tree $\mathcal{T}_\mathrm{a}$ using $g_{\mathrm{a},m}$ and g_m in the first level;
Form a DWT filterbank tree $\mathcal{T}_\mathrm{b}$ using $g_{\mathrm{b},m}$ and g_m delayed by a 1 sample in the first level;
$\mathbf{c}_{\mathrm{a},k} \leftarrow$ Alg. 4 with $\mathcal{T}_\mathrm{a}$; $\mathbf{c}_{\mathrm{b},k} \leftarrow$ Alg. 4 with $\mathcal{T}_\mathrm{b}$;
for $k \leftarrow 0$ **to** $K-1$ **do**
    $\mathbf{c}_k \leftarrow \frac{1}{2}\left(\mathbf{c}_{\mathrm{a},k} + \mathrm{i}\mathbf{c}_{\mathrm{b},k}\right)$; $\mathbf{c}_{2K-1-k} \leftarrow \frac{1}{2}\left(\mathbf{c}_{\mathrm{a},k} - \mathrm{i}\mathbf{c}_{\mathrm{b},k}\right)$;
    % Subbands $\mathbf{c}_k$ and $\mathbf{c}_{2K-1-k}$ are complex conjugates if $\mathbf{f} \in \mathbb{R}^L$
end

ALGORITHM 14: Dual-Tree complex wavelet synthesis

Input: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \ldots, 2K-1\}$ where $K = J(M-1)+1$, Hilbert transform pair of basic filterbanks $g_{\mathrm{a},m}$ and $g_{\mathrm{b},m}$, a_m, first level filterbank g_m, $m \in \{0, 1, \ldots, M-1\}$, number of levels J.
Output: Reconstructed signal $\widetilde{\mathbf{f}} \in \mathbb{C}^{L_s}$.

Form a DWT filterbank tree $\mathcal{T}_\mathrm{a}$ using $g_{\mathrm{a},m}$ and g_m in the first level;
Form a DWT filterbank tree $\mathcal{T}_\mathrm{b}$ using $g_{\mathrm{b},m}$ and g_m delayed by a 1 sample in the first level;
for $k \leftarrow 0$ **to** $K-1$ **do**
    $\mathbf{c}_{\mathrm{a},k} \leftarrow \frac{1}{2}\left(\mathbf{c}_k + \mathbf{c}_{2K-1-k}\right)$; $\mathbf{c}_{\mathrm{b},k} \leftarrow \mathrm{i}\frac{1}{2}\left(\mathbf{c}_{2K-1-k} - \mathbf{c}_k\right)$;
    % $\mathbf{c}_{\mathrm{a},k} \leftarrow \mathrm{real}\left(\mathbf{c}_k\right)$; $\mathbf{c}_{\mathrm{b},k} \leftarrow \mathrm{imag}\left(\mathbf{c}_k\right)$ if $\mathbf{f} \in \mathbb{R}^L$
end
$\widetilde{\mathbf{f}}_\mathrm{a} \leftarrow$ Alg. 5 with $\mathcal{T}_\mathrm{a}$ and $\mathbf{c}_{\mathrm{a},k}$; $\widetilde{\mathbf{f}}_\mathrm{b} \leftarrow$ Alg. 5 with $\mathcal{T}_\mathrm{b}$ and $\mathbf{c}_{\mathrm{b},k}$; $\widetilde{\mathbf{f}} \leftarrow \widetilde{\mathbf{f}}_\mathrm{a} + \widetilde{\mathbf{f}}_\mathrm{b}$;

Toolbox Conventions. Functions implementing Alg. 13 are called `dtwfb` and `dtwfbreal` for real-only signals. Similarly, there are functions `idtwfb` and `idtwfbreal` implementing Alg. 14.

Hilbert transform filterbank pairs suitable for the second and higher levels of trees are stored in functions with `wfiltdt_` prefix: odd and even length biorthogonal filterbanks `oddeven` [Kingsbury 2001], quarter-shift orthogonal filterbanks `qshift` [Kingsbury 2000; 2003], optimized symmetric `optsym` [Dumitrescu et al. 2008] and double-density dual-tree filterbanks `dden` [Selesnick 2004].

6. GENERAL DUAL-TREE COMPLEX FILTERBANK

A generalization of a dual-tree complex wavelet transform analogue to the generalization of DWT described in sec. 3 was proposed in [Bayram and Selesnick 2008]. It is not straightforward as it requires the high-pass filters in the basic filterbanks to meet the half-sample shift condition as well and it further requires another set of *extension* basic wavelet filters to be used in specific nodes in the tree in order to preserve the analytic behaviour. The extension filterbank is shared by both the trees and any basic wavelet filterbank can be used provided it has equal number of filters and downsampling factors as the other filterbanks. Algorithms for analysis and synthesis are Alg. 15 and Alg. 16 respectively.

ALGORITHM 15: General dual-tree complex wavelet filterbank analysis

Input: Input signal $\mathbf{f} \in \mathbb{C}^{L_s}$, Hilbert transform pair of basic filterbanks $g_{\mathrm{a},m}$ and $g_{\mathrm{b},m}$, a_m, first level filterbank $g_{\mathrm{root},m}$ and extension filterbank $g_{\mathrm{ext},m}$, $m \in \{0, 1, \dots, M-1\}$, shape of a tree $\mathcal{T}_{\mathrm{shape}}$.

Output: Complex coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \dots, 2K-1\}$ where K is the total number of unconnected outputs of one of the trees.

Form a filterbank tree $\mathcal{T}_{\mathrm{a}}$ of a shape $\mathcal{T}_{\mathrm{shape}}$ using $g_{\mathrm{root},m}$ in the root node and using $g_{\mathrm{a},m}$ in its direct descendants and in their further lowpass filter connections and using $g_{\mathrm{ext},m}$ elsewhere;

Create $\mathcal{T}_{\mathrm{b}}$ in the same manner using $\mathcal{T}_{\mathrm{b}}$ and one sample delayed version of $g_{\mathrm{root},m}$;

Continue as in Alg. 13 starting with line 3;

ALGORITHM 16: General dual-tree complex wavelet filterbank synthesis

Input: Coefficients $\mathbf{c}_k$ for $k \in \{0, 1, \dots, 2K-1\}$ where K is the total number of unconnected outputs of one of the trees, Hilbert transform pair of basic filterbanks $g_{\mathrm{a},m}$ and $g_{\mathrm{b},m}$, a_m, first level filterbank $g_{\mathrm{root},m}$ and extension filterbank $g_{\mathrm{ext},m}$ $m \in \{0, 1, \dots, M-1\}$, shape of a tree $\mathcal{T}_{\mathrm{shape}}$.

Output: Reconstructed signal $\widetilde{\mathbf{f}} \in \mathbb{C}^{L_s}$.

Form a filterbank tree $\mathcal{T}_{\mathrm{a}}$ of a shape $\mathcal{T}_{\mathrm{shape}}$ using $g_{\mathrm{root},m}$ in the root node and using $g_{\mathrm{a},m}$ in its direct descendants and in their further lowpass filter connections and using $g_{\mathrm{ext},m}$ elsewhere;

Create $\mathcal{T}_{\mathrm{b}}$ in the same manner using $\mathcal{T}_{\mathrm{b}}$ and one sample delayed version of $g_{\mathrm{root},m}$;

Continue as in Alg. 14 starting with line 3;

Toolbox conventions:. General dual-tree complex filterbanks are supported directly by functions `dtwfb*` and `idtwfb*`. The available predefined shapes are as in sec. 3 and the tree structure can be created using `dtwfbinit` and pruned by `wfbtremove`. Adding nodes using `wfbtput` is not yet supported.

7. FRAME BOUNDS OF GENERAL FILTERBANK TREES

This section describes a computationally tractable algorithm for calculating optimal frame bounds A, B of a wavelet frame (any type described in this paper) for large L while avoiding explicit computation of singular values of $\mathbf{F}$ (or equivalently eigenvalues of $\mathbf{F}\mathbf{F}^*$).

The ratio $\sqrt{B/A}$ gives the condition number of a matrix $\mathbf{F}$ with frame vectors as columns, which determines the numeric stability of the system, and B alone corresponds to the spectral (operator) norm of such matrix as $\|\mathbf{F}\| = \sqrt{B}$. Performance of many iterative algorithms depends on the knowledge of frame bounds and/or they are restricted to be working with matrices $\|\mathbf{F}\| \leq 1$ only (e.g. the frame algorithm [Gröchenig 1993], iterative soft thresholding [Daubechies et al. 2004] and proximal convex optimization algorithms in general [Combettes and Pesquet 2011]).

The literature on the frame bounds of iterated filterbanks is sparse, and only the DWT case in $\ell^2(\mathbb{Z})$ was addressed as the analytical treatment of general filterbank trees would become cumbersome. In [Stanhill and Zeevi 1996], the authors established some connection between the bounds of the basic wavelet filterbank and the J-level DWT. Namely the following holds for a basic filterbank with bounds $A_1, B_1 \geq 1$

$$A_J \geq A_1 \text{ and } B_J \leq B_1^J, \tag{5}$$

where the subscripts denote number of levels in the DWT filterbank. For tight basic filterbanks with bounds equal to $A_1 = B_1 > 1$ the following was proven

$$A_J = A_1 \text{ and } B_J \leq A_1^J, \tag{6}$$

which shows that the tightness might be lost during iterations. The exception is the Parseval tightness $A_1 = 1$, which is preserved under iterations $A_J = B_J = 1$. Results (5),(6) also hold for the undecimated DWT. It has been observed in [Fowler 2005] that the bounds of the J-level undecimated DWT with (originally orthonormal) two-channel basic filterbank with $A_1 = B_1 = 2$ (`noscale` flag from sec. 4) are $A_J = 2$, $B_J = 2^J$ exactly.

We observed the same behaviour for bounds for frames in $\mathbb{C}^L$, calculated using algorithms described in this section.

Although the precise relationship between the frame bounds in $\mathbb{C}^L$ and $\ell^2(\mathbb{Z})$ has not yet been addressed in the literature, in the practical applications conducted in $\mathbb{C}^L$, we are usually interested in bounds of the frame at hand.

The numerical evaluation of frame bounds of a general filterbank tree is not straightforward and it requires two pre-processing steps:

(1) Using properties from sec. 1.4, transform a filterbank tree of interest to an identical, possibly non-uniform filterbank.
(2) Transform the resulting filterbank to an uniform one using the identity by [Akkarakaran and Vaidyanathan 2003] (summarized in Alg. 17).

Having an equivalent uniform filterbank frame, the optimal frame bounds in $\mathbb{C}^L$ can be obtained such that the lower frame bound is the the minimum of lower frame bounds of a length-$\frac{L}{a}$ series of smaller frames (see below) in $\mathbb{C}^a$ and, likewise, the upper frame bound is the maximum of the upper frame bounds of such frames [Fickus et al. 2013]. Although not specifically mentioned in their paper, such a behaviour can be explained trough the analysis of the frame operator which exhibit a regular behaviour in both the time and the frequency domains. In the time domain, the frame operator is a block-circulant matrix and in the frequency domain, it can become a block-diagonal matrix consisting of $\frac{L}{a}$ blocks of size $a \times a$ after suitable row and column permutations. Eigenvalues of a block-diagonal matrix are just the concatenation of eigenvalues of separate block-matrices and therefore the frame bounds are easily obtained. For sufficiently small a, the blocks can be created explicitly and the eigenvalues of blocks evaluated directly as shown in Alg. 18.

The block-diagonal form of the frame operator can be also exploited for effective computation of canonical dual frames of uniform filterbanks and even nonuniform ones with $\mathrm{lcm}\left(a_0, a_1, \ldots, a_{M-1}\right)$ being resonably low. We omit the description of the algorithm as it is not the main objective of the paper and we refer the interested reader to function `filterbankdual` and `nonu2ufilterbank`.

ALGORITHM 17: Non-uniform to uniform filterbank transformation

---

Input: Nonuniform filterbank g_m, a_m for $m \in \{0, 1, \ldots, M-1\}$.
Output: Uniform filterbank h_k, a for $k \in \{0, 1, \ldots, K-1\}$, where
$K = \sum_{m=0}^{M-1} \mathrm{lcm}\left(a_0, a_1, \ldots, a_{M-1}\right)/a_m$.
$a = \mathrm{lcm}\left(a_0, a_1, \ldots, a_{M-1}\right)$;
$k = 0$;
for $m \leftarrow 0$ **to** $M-1$ **do**
  for $p \leftarrow 0$ **to** $a/a_m - 1$ **do**
    $h_k \leftarrow g_m$ with $d_{h_k} = d_{g_m} - a_m p$; `% Adjust filter ‘‘offset’’`
    $k \leftarrow k+1$;
  end
end

ALGORITHM 18: Frame bounds of an uniform filterbank in $\mathbb{C}^L$

Input: Uniform filterbank $\mathbf{g}_m \in \mathbb{C}^L$ and a for $m \in \{0, 1, \ldots, M-1\}$, system length $L = Na$ for some $N \in \mathbb{N}$.
Output: Optimal frame bounds A, B.
$A \leftarrow \infty, B \leftarrow 0$;
$\mathbf{G} \leftarrow$ stack $\widehat{\mathbf{g}_m}$ as columns ; `%` $\mathbf{G} \in \mathbb{C}^{L \times M}$
for $w \leftarrow 0$ **to** $N-1$ **do**
  $\mathbf{H} \leftarrow$ stack rows $(w + kN)$ of $\mathbf{G}$ for $k \in \{0, 1, \ldots, a-1\}$; `%` $\mathbf{H} \in \mathbb{C}^{a \times M}$
  $\mathbf{e} \leftarrow \mathrm{eig}(\mathbf{H}\mathbf{H}^*)$; `% Compute eigenvalues of a` $a \times a$ `matrix`
  $A \leftarrow \min(A, \min(\mathbf{e}))$; $B \leftarrow \max(B, \max(\mathbf{e}))$;
end
$A \leftarrow A/a$; $B \leftarrow B/a$;

Alg. 18 can be simplified for undecimated transforms ($a = 1$) while the frame operator in the Fourier domain is diagonal (Alg. 19) (and even constant in the tight frame case).

ALGORITHM 19: Frame bounds of an undecimated filterbank in $\mathbb{C}^L$

Input: Undecimated filterbank $\mathbf{g}_m \in \mathbb{C}^L$ for $m \in \{0, 1, \ldots, M-1\}$, system length $L \in \mathbb{N}^+$.
Output: Optimal frame bounds A, B.
$\mathbf{h} \leftarrow \sum_{m=0}^{M-1} |\widehat{\mathbf{g}_m}|^2$; `% Square and the abs. operation are performed element-wise.`
$A \leftarrow \min(\mathbf{h})$; $B \leftarrow \max(\mathbf{h})$;

Toolbox conventions. Toolbox functions for calculating frame bounds are called `wfbtbounds` and `wpfbtbounds` for both the decimated case and for the undecimated transforms. Note there are no such functions for `fwt` and `ufwt` as they are both special cases of `wfbt` and `uwfbt` respectively. The function calculating the frame bounds of the dual-tree wavelet transform from sec. 5 is called `dtwfbbounds`. All functions require defining compatible L, equal or longer than the longest identical filter according to Alg. 1 and flags specific to the respective transforms.

8. CONCLUSION AND OUTLOOK

The hope of the authors is that the toolbox becomes a base for new numerical experiments and scientific developments.

In the future, apart from the topics already mentioned in the introduction, the toolbox might be enriched by inclusion of the following:

Support of a systematic construction of tight wavelet frames and bi-framelets [Ron and Shen 1997; Benedetto and Li 1998; Daubechies et al. 2003; Ehler 2007] trough unitary, oblique and mixed extension principles, which is an active area of research even nowadays. As the fast algorithms mostly coincide with the already implemented fast wavelet transform with more than two wavelet filters, it is expected that the results will fit into the already existing framework seamlessly.

Inclusion of more general wavelet frames (not based on MRA) via frequency warping [Holighaus et al. 2015; Christensen and Goh 2014]. These recent results might be the answer to the ongoing quest for the truly discrete wavelet frames with a denser logarithmic frequency resolution.

APPENDIX

This section contains Alg. 20 and Fig. 5 which were moved here to avoid cluttering Section 3 they relate to. Alg. 20 is used as a subprocess in Alg. 4 and Alg. 5.

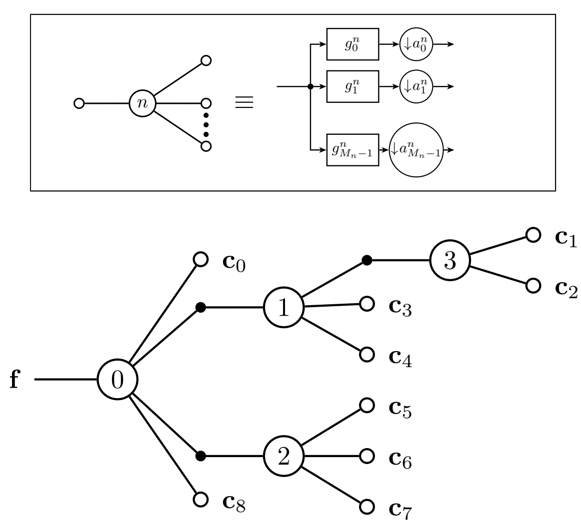

Fig. 5. An example of a schematic diagram of a filterbank tree as used in Alg. 4, Alg. 5 and Alg. 20. The tree consists of $N = 4$ nodes, the number of filters M_n in node n is $M_0 = 4$, $M_1 = 3$, $M_2 = 3$, $M_3 = 2$, where n (the number in the nodes) is the index in the breadth-first order. There are $K = 9$ unconnected outputs of the tree shown as empty circles. The output of Alg. 20 for this particular tree shape is $kIdMap = \{0, 8, 3, 4, 5, 6, 7, 1, 2\}$.

ALGORITHM 20: Compute filterbank tree output index map

Input: Filterbank tree $\mathcal{T}$ with N nodes in the breadth-first order where the n-th node consists of M_n filters and $M_{\mathrm{unc},n}$ of them are unconnected.
Output: Array of indices *kIdMap* of length $K = \sum_{n=0}^{N-1} M_{\mathrm{unc},n}$ (the total number of unconnected outputs of all nodes).

$k \leftarrow 0$;
Prepare array *kIdMap* of length K ;
nodeFiltTuple $\leftarrow \{\}$; `% an empty LIFO stack of index tuple` $(n, m_{\mathrm{start}}, m_{\mathrm{unc}})$
push $(0, 0, 0)$ to *nodeFiltTuple* ;
while *nodeFiltTuple is not empty* **do**
    $(n, m_{\mathrm{start}}, m_{\mathrm{unc}}) \leftarrow$ pop from *nodeFiltTuple* ;
    for $m \leftarrow m_{\mathrm{start}}$ **to** $M_n - 1$ **do**
        if *node* n *has a next node* n_{next} *connected at index* m **then**
            if $m < M_n - 1$ **then**
                push $(n, m+1, m_{\mathrm{unc}})$ to *nodeFiltTuple* ;
            end
            push $(n_{\mathrm{next}}, 0, 0)$ to *nodeFiltTuple* ;
            break ;
        else
            $kIdMap(m_{\mathrm{unc}} + \sum_{l=0}^{n-1} M_{\mathrm{unc},l}) = k$;
            $k \leftarrow k + 1$; $m_{\mathrm{unc}} \leftarrow m_{\mathrm{unc}} + 1$;
        end
    end
end

ACKNOWLEDGMENTS

Authors would like to acknowledge work of authors of the Uvi_Wave toolbox [Prelcic et al. 1996] from which some of the wavelet filter generation routines were taken over.

REFERENCES

A. Farras Abdelnour. 2007. Dense grid framelets with symmetric lowpass and bandpass filters, In Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on. *Applied mathematics and computation* 172 (2007), 1–4.

A. Farras Abdelnour. 2012. Symmetric wavelets dyadic sibling and dual frames. *Signal Processing* 92, 5 (2012), 1216 – 1229.

A. Farras Abdelnour and Ivan W. Selesnick. 2004. Symmetric nearly orthogonal and orthogonal nearly symmetric wavelets. *The Arabian Journal for Science and Engineering* 29, 2C (2004), 3 – 16.

A. Farras Abdelnour and Ivan W. Selesnick. 2005. Symmetric nearly shift-invariant tight frame wavelets. *IEEE Transactions on Signal Processing* 53, 1 (2005), 231–239.

Sony Akkarakaran and P. P. Vaidyanathan. 2003. Nonuniform filter banks: New results and open problems. In *Studies in Computational Mathematics: Beyond Wavelets*, P. Monk C.K. Chui and L. Wuytack (Eds.). Vol. 10. Elsevier B.V., 259 –301.

O. Alkin and H. Caglar. 1995. Design of efficient M-band coders with linear-phase and perfect-reconstruction properties. *Signal Processing, IEEE Transactions on* 43, 7 (jul 1995), 1579 –1590.

Peter Balazs. 2007. Basic definition and properties of Bessel multipliers. *J. Math. Anal. Appl.* 325, 1 (2007), 571 – 585.

Peter Balazs. 2008. Frames and Finite Dimensionality: Frame Transformation, Classification and Algorithms. *Applied Mathematical Sciences* 2, 41–44 (2008), 2131–2144.

Peter Balazs, Monika Dörfler, Florent Jaillet, Nicki Holighaus, and Gino Angelo Velasco. 2011. Theory, implementation and applications of nonstationary Gabor frames. *J. Comput. Appl. Math.* 236, 6 (2011), 1481–1496.

Ilker Bayram and Ivan. W. Selesnick. 2008. On the Dual-Tree Complex Wavelet Packet and M-Band Transforms. *Signal Processing, IEEE Transactions on* 56, 6 (June 2008), 2298–2310.

John J. Benedetto and Shidong Li. 1998. The Theory of Multiresolution Analysis Frames and Applications to Filter Banks. *Applied and Computational Harmonic Analysis* 5, 4 (1998), 389 – 427.

Helmut Bölcskei, Franz Hlawatsch, and Hans G. Feichtinger. 2002. Frame-theoretic analysis of oversampled filter banks. *Signal Processing, IEEE Transactions on* 46, 12 (2002), 3256–3268.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. 2011. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* 3, 1 (Jan. 2011), 1–122.

C. Sidney Burrus, Ramesh Gopinath, and Haitao Guo. 2013. Wavelets and Wavelet Transforms. *OpenStax_CNX* (Feb. 2013). http://cnx.org/content/col11454/1.5/

Peter G. Casazza and Gitta Kutyniok (Eds.). 2013. *Finite Frames: Theory and Applications*. Birkhäuser, Basel, Switzerland.

Ole Christensen and Say Song Goh. 2014. From dual pairs of Gabor frames to dual pairs of wavelet frames and vice versa. *Applied and Computational Harmonic Analysis* 36, 2 (2014), 198–214.

Ronald R. Coifman and Mladen Victor Wickerhauser. 1992. Entropy-based algorithms for best basis selection. *Information Theory, IEEE Transactions on* 38, 2 (March 1992), 713–718.

Patrick L. Combettes and Jean-Christophe Pesquet. 2011. Proximal Splitting Methods in Signal Processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, Heinz H. Bauschke, Regina S. Burachik, Patrick L. Combettes, Veit Elser, D. Russell Luke, and Henry Wolkowicz (Eds.). Springer New York, 185–212.

Z. Cvetkovic and M. Vetterli. 1998. Oversampled filter banks. *Signal Processing, IEEE Transactions on* 46, 5 (May 1998), 1245–1255.

Ingrid Daubechies. 1992. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Ingrid Daubechies, Michel Defrise, and Christine De Mol. 2004. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. Pure Appl. Math.* 57, 11 (2004), 1413–1457.

Ingrid Daubechies, A. Grossmann, and Y. Meyer. 1986. Painless nonorthogonal expansions. *J. Math. Phys.* 27, 5 (1986), 1271–1283.

Ingrid Daubechies, Bin Han, Amos Ron, and Zuowei Shen. 2003. Framelets: MRA-based constructions of wavelet frames. *Applied and Computational Harmonic Analysis* 14, 1 (2003), 1–46.

Bogdan Dumitrescu, Ilker Bayram, and Ivan W. Selesnick. 2008. Optimization of Symmetric Self-Hilbertian Filters for the Dual-Tree Complex Wavelet Transform. *IEEE Signal Process. Lett.* 15 (2008), 146–149.

Martin Ehler. 2007. *The construction of nonseparable wavelet bi-frames and associated approximation schemes*. Ph.D. Dissertation. Philipps–Universität Marburg, Marburg/Lahn, Germany.

Matthew Fickus, Melody L. Massar, and Dustin G. Mixon. 2013. Finite Frames and Filter Banks. In *Finite Frames: Theory and Applications*, Peter G. Casazza and Gitta Kutyniok (Eds.). Birkhäuser, Basel, Switzerland, 337–380.

James E. Fowler. 2005. The Redundant Discrete Wavelet Transform and Additive Noise. *Signal Processing Letters, IEEE* 12, 9 (2005), 629–632.

Ramesh A. Gopinath and C. Sidney Burrus. 1995. On cosine-modulated wavelet orthonormal bases. *Image Processing, IEEE Transactions on* 4, 2 (Feb 1995), 162–176.

D. Griffin and J. Lim. 1984. Signal estimation from modified short-time Fourier transform. *IEEE Trans. Acoust. Speech Signal Process.* 32, 2 (1984), 236–243.

Karlheinz Gröchenig. 1993. Acceleration of the frame algorithm. *Signal Processing, IEEE Transactions on* 41, 12 (Dec 1993), 3331–3340.

Nicki Holighaus, Zdeněk Průša, and Christoph Wiesmeyr. 2015. Designing tight filter bank frames for nonlinear frequency scales. *Sampling Theory and Applications 2015* (2015). submitted.

M. Holschneider, R. Kronland-Martinet, J. Morlet, and Ph. Tchamitchian. 1990. A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform. In *Wavelets*, Jean-Michel Combes, Alexander Grossmann, and Philippe Tchamitchian (Eds.). Springer Berlin Heidelberg, 286–297.

Nick G. Kingsbury. 2000. A Dual-Tree Complex Wavelet Transform with Improved Orthogonality and Symmetry Properties.. In *ICIP*. 375–378.

Nick G. Kingsbury. 2001. Complex Wavelets for Shift Invariant Analysis and Filtering of Signals. *Applied and Computational Harmonic Analysis* 10, 3 (2001), 234 – 253.

Nick G. Kingsbury. 2003. Design of Q-shift complex wavelets for image processing using frequency domain energy minimization. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, Vol. 1. I–1013–16 vol.1.

Jelena Kovačević and Amina Chebira. 2008. An Introduction to Frames. *Found. Trends Sig. Proc.* 2, 1 (Oct. 2008), 1–94.

Tony Lin, Shufang Xu, Qingyun Shi, and Pengwei Hao. 2006. An algebraic construction of orthonormal M-band wavelets with perfect reconstruction. *Applied mathematics and computation* 172, 2 (2006), 717–730.

Stéphane G. Mallat. 1989. A Theory for Multiresolution Signal Decomposition: The Wavelet Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* 11, 7 (July 1989), 674–693.

Nuria González Prelcic, Oscar W. Márquez, and Santiago González. 1996. Uvi Wave, the ultimate toolbox for wavelet transforms and filter banks. In *Proceedings of the Fourth Bayona Workshop on Intelligent Methods in Signal Processing and Communications*. Bayona, Spain, 224–227.

Zdeněk Průša, Peter L. Søndergaard, Nicki Holighaus, Christoph Wiesmeyr, and Peter Balazs. 2014. The Large Time-Frequency Analysis Toolbox 2.0. In *Sound, Music, and Motion*, Mitsuko Aramaki, Olivier Derrien, Richard Kronland-Martinet, and Sølvi Ystad (Eds.). Springer International Publishing, 419–442.

Pavel Rajmic and Zdeněk Průša. 2014. Discrete Wavelet Transform of Finite Signals: Detailed Study of the Algorithm. *International Journal of Wavelets, Multiresolution and Information Processing* 12, 01 (2014), 1450001.

Olivier Rioul. 1993. A discrete-time multiresolution theory. *IEEE Transactions on Signal Processing* 41, 8 (1993), 2591–2606.

Amos Ron and Zuowei Shen. 1997. Affine Systems in $L^2(\mathbb{R}^d)$: The Analysis of the Analysis Operator . *Journal of Functional Analysis* 148, 2 (1997), 408 – 447.

Ivan W. Selesnick. 2001. The double density DWT. In *Wavelets in Signal and Image Analysis*. Springer, 39–66.

Ivan W. Selesnick. 2004. The double-density dual-tree DWT. *Signal Processing, IEEE Transactions on* 52, 5 (May 2004), 1304–1314.

Ivan W. Selesnick. 2006. A higher density discrete wavelet transform. *IEEE Transactions on Signal Processing* 54, 8 (2006), 3039–3048.

Ivan W. Selesnick and A. Farras Abdelnour. 2004. Symmetric wavelet tight frames with two generators. *Appl. Comput. Harmon. Anal.* 17, 2 (2004), 211–225.

Ivan W. Selesnick, Richard G. Baraniuk, and Nick G. Kingsbury. 2005. The dual-tree complex wavelet transform. *Signal Processing Magazine, IEEE* 22, 6 (nov. 2005), 123 – 151.

M. Shensa. 1992. The Discrete Wavelet Transform: Wedding the À Trous and Mallat Algorithms. *IEEE Transactions on Signal Processing* 40, 10 (Oct 1992), 2464–2482.

Peter L. Søndergaard, Bruno Torrésani, and Peter Balazs. 2012. The Linear Time Frequency Analysis Toolbox. *International Journal of Wavelets, Multiresolution Analysis and Information Processing* 10, 4 (2012).

David Stanhill and Yehoshua Y. Zeevi. 1996. Frame analysis of wavelet type filter banks. In *Digital Signal Processing Workshop Proceedings, 1996., IEEE*. 435–438.

Peter Steffen, Peter N. Heller, Ramesh A. Gopinath, and C. Sidney Burrus. 1993. Theory of regular M-band wavelet bases. *Signal Processing, IEEE Transactions on* 41, 12 (Dec 1993), 3497–3511.

Gilbert Strang and T. Nguyen. 1997. *Wavelets and filter banks*. Wellesley-Cambridge Press.

Wim Sweldens. 1996. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comput. Harmon. Anal.* 3, 2 (1996), 186–200.

Carl Taswell. 1994. Near-Best Basis Selection Algorithms With Non-Additive Information Cost Functions. In *Proceedings of the IEEE International Symposium on Time-Frequency and Time-Scale Analysis*. IEEE Press, 13–16.

Carl Taswell and Kevin C. McGill. 1994. Algorithm 735: Wavelet Transform Algorithms for Finite-duration Discrete-time Signals. *ACM Trans. Math. Softw.* 20, 3 (Sept. 1994), 398–412.

P. P. Vaidyanathan. 1993. *Multirate Systems and Filter Banks*. Prentise-Hall, Englewood Clifs, NJ.

Martin Vetterli, Jelena Kovačević, and Vivek K Goyal. 2014. *Foundations of Signal Processing*. Cambridge Univ. Press. http://www.fourierandwavelets.org/

Mladen Victor Wickerhauser. 1994. *Adapted wavelet analysis from theory to software.* Wellesley-Cambridge Press, Wellesley, MA.

Received February 2007; revised March 2009; accepted June 2009