The ~~Linear~~ Large Time-Frequency Analysis Toolbox: Wavelets

Zdeněk Průša

Acoustics Research Institute, Austrian Academy of Sciences

OctConf 2013, June 25, 2013

Overview

- Wavelets in the LTFAT.
- Real-time audio block-stream processing framework.
- Example of a real-time audio wavelet processing in Octave.

Introduction

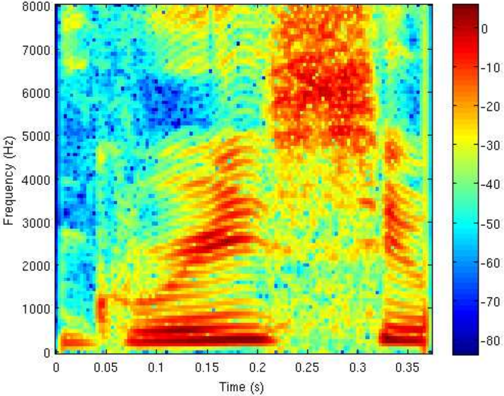
LTFAT is a modern Octave/Matlab toolbox for doing time/frequency, wavelet and frame analysis.

Started in 2004 by Peter L. Søndergaard. Version 1.0 released in 2011.

Its purposes are:

- To support teaching and learning in Fourier analysis, harmonic analysis and digital signal processing.
- To provide a tested and documented toolbox of such quality that it can be used for new scientific developments.
- As a method for engineers and researchers to quickly try out a method/transform.
- As a method for researchers to push their discoveries to a larger audience.

Time-frequency representation example



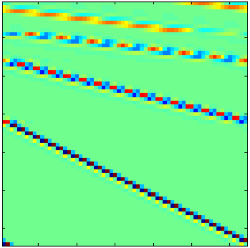
Features

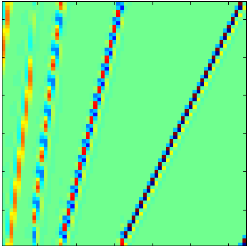
- Basic Fourier analysis and signal processing, FIR windows
- Discrete Gabor transform and its inverse
- Time-frequency bases: Wilson and WMDCT
- Filterbanks and non-stationary Gabor systems
- Reassignment (sharpening) and instantaneous frequency estimation
- Non-linear analysis and synthesis methods
- Backend in C linked to OCT interfaces.
- (NEW) Discrete Wavelet Transform
- (NEW) Block-stream processing framework
- . . .

Wavelets in the LTFAT overview

- fwt – Discrete Wavelet Transform (Mallat's algorithm)
- ufwt – Undecimated fwt (À-trous algorithm).
- wfbt/uwfbt – (Undecimated) Arbitrary tree-shaped Wavelet filterbank.
- wpfbt/uwpfbt – (Undecimated) Arbitrary tree-shaped Wavelet filterbank.
- wpbest – Best basis selection from bases derived from the wavelet packet.
- fwt2 – Basic 2D Discrete wavelet transform.
- Wavelet filters library.
- Plotting routines.

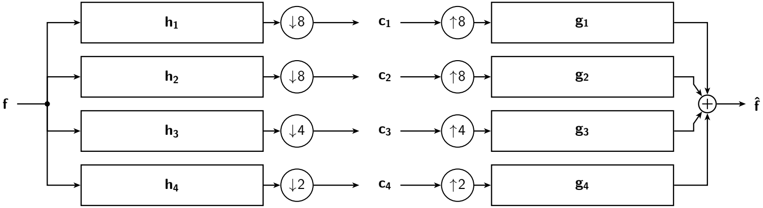
DWT as a matrix multiplication

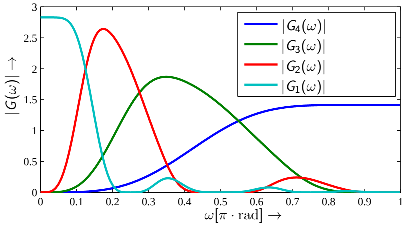
$c =$  f

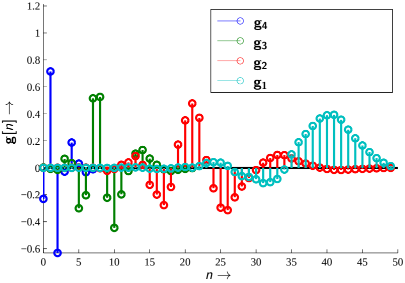
$\hat{f} =$  c

Daubechies 4, 3 scale levels, $N = 64$

DWT as a filterbank

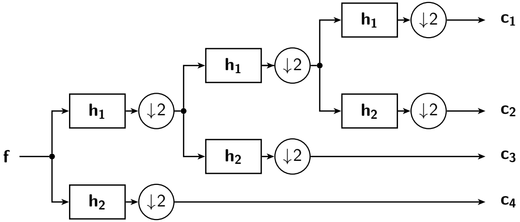


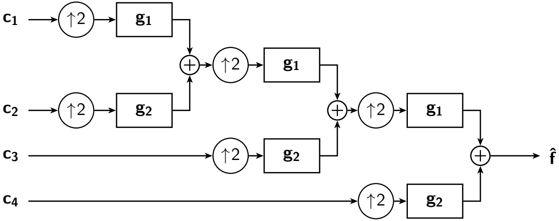




Daubechies 4, 3 scale levels.

Fast Wavelet Transform





Mallat's fast algorithm, $\mathbf{h_1}/\mathbf{h_2}$ $(\mathbf{g_1}/\mathbf{g_2})$ – lowpass/highpass filters.

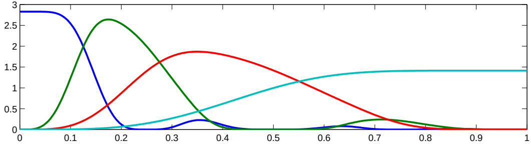
Fast Wavelet Transform – `fwt/ifwt`

Example:

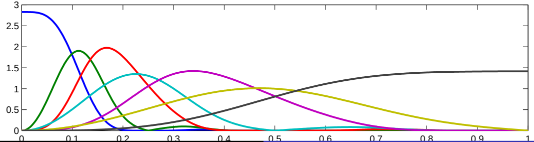
```
c = fwt(f,'db4',3);
fhat = ifwt(c,'db4',3,size(f,1));
```

Other filterbank constructions with different number of filters in the basic filterbank. Offers more convenient filters.

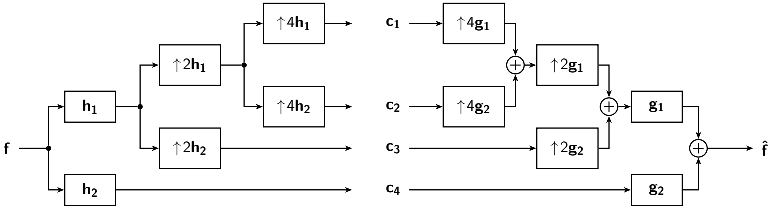
`db4`, 3 levels



`dden2`, 3 levels

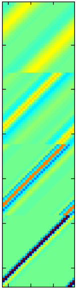
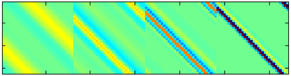


Fast Undecimated Wavelet Transform



À-trous algorithm, $\mathbf{h_1}/\mathbf{h_2}$ ($\mathbf{g_1}/\mathbf{g_2}$) – lowpass/highpass filters, $\uparrow N$ – upsampling by factor of N

Undec. Wavelet Transform – `ufwt/iufwt`

$c =$  f, $\hat{f} =$  c

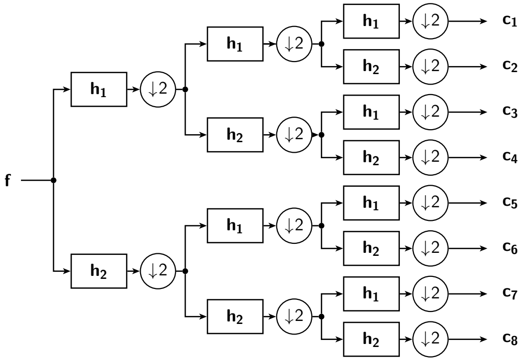
Example:

```
c = ufwt(f,'db4',3);
fhat = iufwt(c,'db4',3);
```

Highly redundant, shift-independent transform.

To fwt

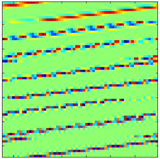
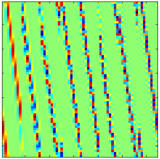
Arbitrary Wavelet filterbank – wfbt/iwfbt



Example of a full 3 level tree:

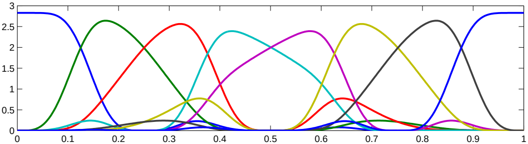
```
c = wfbt(f,{'db4',3,'full'});
fhat = iwfbt(c,{'db4',3,'full'},size(f,1));
```

Arbitrary Wavelet filterbank – `wfbt/iwfbt`

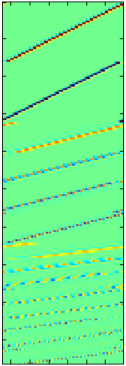
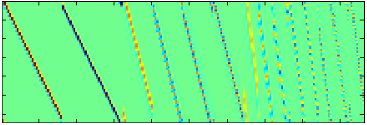
$c =$  $f, \qquad \hat{f} =$  c

Allows flexible frequency covering via splitting further the high-pass output.

`db4`, 3 levels



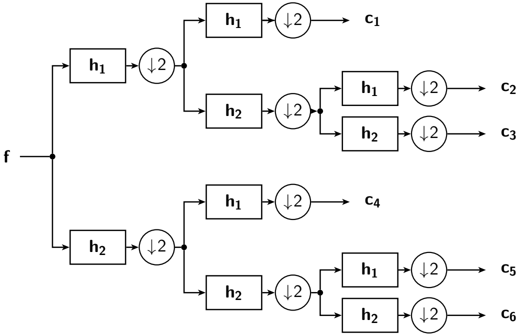
Wavelet Packet filterbank – wpfbt/iwpfbt

$c =$  $f,$ $\hat{f} =$  c

Example of a full 3 level tree:

```
c = wpfbt(f,{'db4',3,'full'});
fhat = iwpfbt(c,{'db4',3,'full'},size(f,1));
```

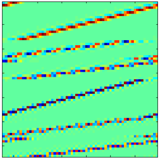
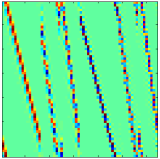
Best subtree – wpbest



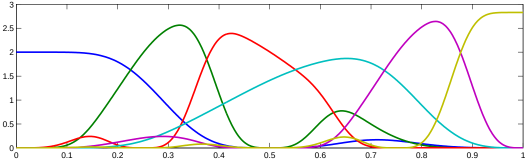
Example

```
[c,wt] = wpbest(f,'db4',3,'entropy','shannon');
fhat = iwfbt(c,wt,size(f,1));
```

Best subtree – wpbest

$c =$  $f, \qquad \hat{f} =$  c

db4, 3 levels



Real-time audio processing in LTFAT

- Simple framework for a non-blocking real-time audio processing and playback.
- Based on `Playrec` (`http://www.playrec.co.uk/`) MEX interface to `Portaudio` library (`http://www.portaudio.com/`).
- Takes input from a sound file or any audio input (microphone, line-in) and routes to any output device (speakers, line-out) allowing processing sample blocks on-the-fly.

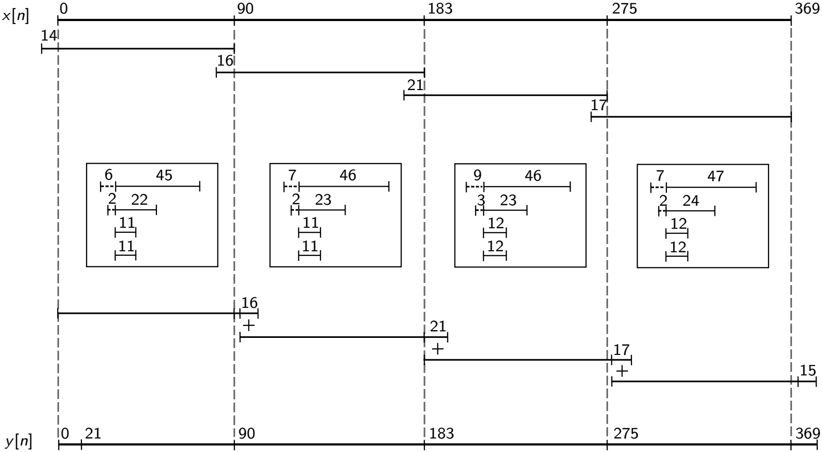
Real-time audio processing in LTFAT cont.

Example:

```
block(’gspi.wav’);  % Input is a wav file.
% block(’playrec’); % Input is an microphone.

% Setup GUI control panel containing one slider.
p = blockpanel({’GdB’,’Gain’,-20,20,0,21});

while p.flag
   % Obtain parameter from a GUI
   gain = 10^(p.getParam(’GdB’)/20);

   % Read 1024 samples from the input
   f = blockread(1024);

   % Enqueue samples to be played
   blockplay(f*gain);
end
p.close();
```

SegDWT (Průša, Rajmic 2012)

Avoiding block-artifacts after coefficient manipulation.



Currently only `fwt`/`ifwt` routines are supported. The plan is to extend the idea of SegDWT to more filterbank types.

SegDWT example

```
block('gspi.wav');  % block('playrec');
F = frame('fwt','sym8',4);
% Setup GUI control panel containing two sliders.
p = blockpanel({{'GdB','Gain',-20,20,0,21},...
                {'Thr','Treshold',0,0.1,0,1000}});
while p.flag
   % Get the current slider value.
   gain = 10^(p.getParam('GdB')/20);
   thres = p.getParam('Thr');

   % Read 1024 samples of the input and process.
   f = blockread(1024);
   c = blockana(F, f*gain);
   c = thresh(c,thres,'soft');
   fhat = blocksyn(F, c, size(f,1));

   % Enqueue the samples to be played.
   blockplay(fhat);
end
p.close();
```

Wavelet-related plans

- Long standing inclusion request from YAWTB `http://sites.uclouvain.be/ispgroup/yawtb/`.
  - (Discretized) Continuous Wavelet Transfom – CWT (Morlet, Mexican hat, . . .).
  - Directional "framed" 2D Wavelet Transform.
  - Wavelet transform on a sphere.
- General Wavelets frames.
- Making LTFAT a proper Octave package ;)

Thank you for listening.

LTFAT

`http://ltfat.sourceforge.net/`