The Linear Time-Frequency Analysis Toolbox: Wavelets

Zdeněk Průša
Acoustics Research Institute, Austrian Academy of Sciences

OctConf 2013, June 25, 2013
Wavelets in the LTFAT.
Real-time audio block-stream processing framework.
Example of a real-time audio wavelet processing in Octave.
LTFAT is a modern Octave/Matlab toolbox for doing time/frequency, wavelet and frame analysis.


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

Zdeněk Průša  http://ltfat.sourceforge.net/
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.
- `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.
Mallat’s fast algorithm, $h_1/h_2$ ($g_1/g_2$) – lowpass/highpass filters.
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
À-trous algorithm, $h_1/h_2$ ($g_1/g_2$) – lowpass/highpass filters, $\uparrow N$ – upsampling by factor of $N$
Undec. Wavelet Transform – ufw/iufw

Example:

\[ c = \text{ufwt}(f, 'db4', 3); \]
\[ \hat{f} = \text{iufwt}(c, 'db4', 3); \]

Highly redundant, shift-independent transform.
Example of a full 3 level tree:

```matlab
c = wfbt(f,{'db4',3,'full'});
fhat = iwfbt(c,{'db4',3,'full'},size(f,1));
```
Arbitrary Wavelet filterbank – wfbt/iwfbt

\[ c = f, \quad \hat{f} = c \]

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

db4, 3 levels
\[ c = \text{wpfbt}(f, \{'db4', 3, 'full'\}); \]
\[ \hat{f} = \text{iwpfbt}(c, \{'db4', 3, 'full'\}, \text{size}(f,1)); \]
Example

\[ [c, wt] = \text{wpbest}(f, 'db4', 3, 'entropy', 'shannon'); \]
\[ \text{fhat} = \text{iwfbt}(c, wt, \text{size}(f, 1)); \]
Best subtree – wpbest

c = f, \hat{f} = c

db4, 3 levels
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.
Example:

```matlab
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();
```
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

```matlab
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
  - (Discretized) Continuous Wavelet Transform – 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.

http://ltfat.sourceforge.net/