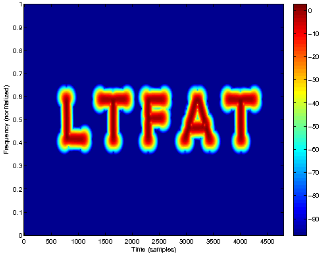
Filterbanks and block-processing in LTFAT

Zdeněk Průša

Acoustics Research Institute (ARI)
Austrian Academy of Sciences, Vienna

3rd SPLab Workshop, 1.11.2013

Introduction



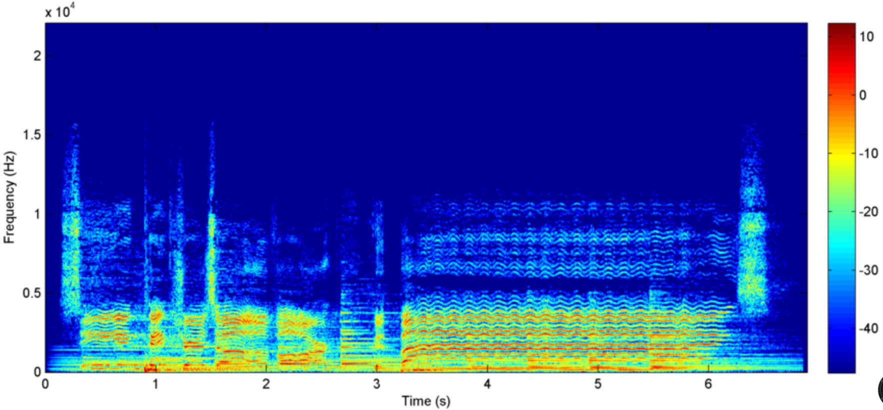
is a Matlab/Octave toolbox for working with time-frequency analysis and synthesis. It is intended both as an educational and a computational tool. The toolbox provides a large number of linear transforms including Gabor and wavelet transforms along with routines for constructing windows (filter prototypes) and routines for manipulating coefficients.

LTFAT Overview

- Started in 2004 by Peter L. Søndergaard, 1.0 released in 2011.
- Tested and well documented – `mat2doc`
- MEX/OCT interfaces to the backend lib in C.
- Build system independent of Matlab's `mex` command.
- Cross-platform, Matlab/Octave, open source, GPL3
- `http://ltfat.sourceforge.net`

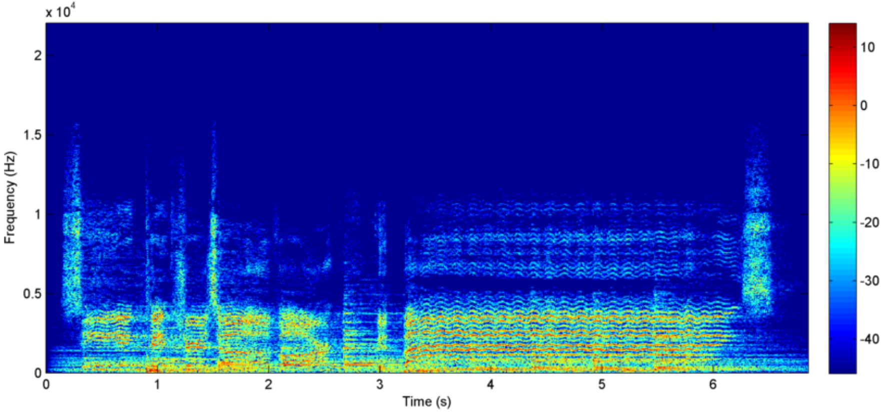
Discrete Gabor Transform $R = 16$

. . . a picture is worth a thousand words . . .



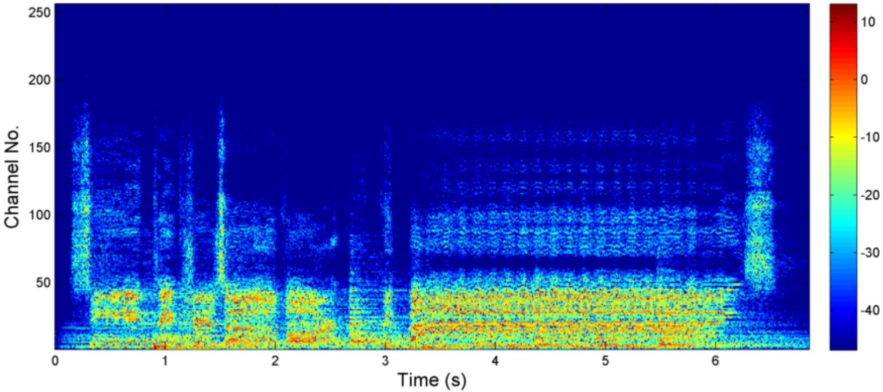
```
F = frame('dgtreal',{'hann',882},60,1000);
plotframe(F,frana(F,f),fs,'dynrange',60);
```

Windowed MDCT $R = 1$



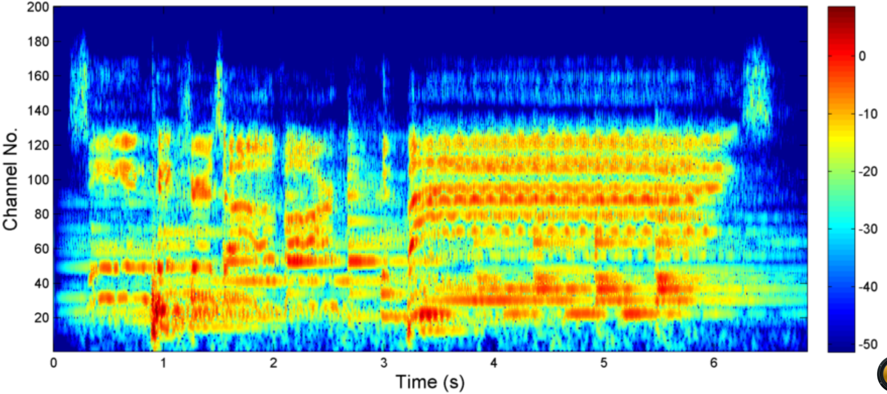
```
F = frame('wmdct',{'hann',882},441);
```

Wavelet Packet subtree $R = 1$



```
F = frame('wfbt',{'sym10',8})
```

Erblets $R \sim 12.6$



```
[g,a]=erbfilters(fs,'fractional','L',numel(f),'M',200,'real')
F = frame('filterbankreal',g,a,numel(g));
```

Talk Outline

1. Current state of LTFAT
2. Filterbanks
3. Block-processing framework
   (and live demonstration)

Current state of LTFAT

Current development version 1.4.2.
Version 2 until end of the year!

Main features in LTFAT 2.0:

- Frames framework
- Wavelets module
- Block-processing framework

Frames framework

– All your frame are belong to us – . . . . P. L. Søndergaard

- The mathematical idea of a "frame" fits well with the notion of class in OOP:
- Each frame has some properties: upper and lower bounds, redundancy, etc.:
  $\Longrightarrow$ object attributes.
- Each frame is always associated with analysis and synthesis operators:
  $\Longrightarrow$ object methods.
- Simple custom object system using structs.
  - Old (pre 2008a) and new OOP in Matlab.
  - Octave compatibility.

Frames framework – overview

– All your frame are belong to us – .... P. L. Søndergaard

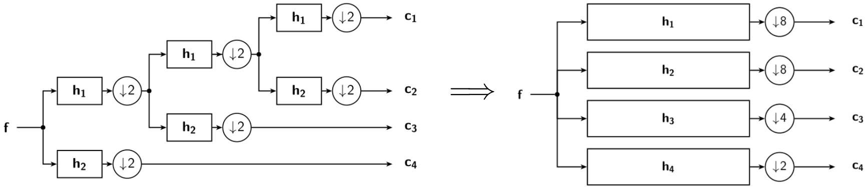
- `F = frame` – create a new freame
- `frana(F,...)` – frame analysis operator
- `frsyn(F,...)` – frame synthesis operator
- `framematrix(F,...)` – matrix form of syntesis operator
- `framedual(F,...)` – construct a dual frame
- `frametight(F,...)` – construct a tight frame
- `franalasso(F,...)` – minimizes $\frac{1}{2}||(f - Fc)||_2^2 + \lambda||c||_1$ (F)ISTA
- `franaiter(F,...)` – iterative analysis using synthesis operator
- `frsyniter(F,...)` – iterative synthesis using analysis opearator
- `frsynabs(F,...)` – synthesis using only abs. values (Griffin-Lim)
- `frameaccel(F,L)` – precompute stuff for given length
- `plotframe(F,...)` – plot frame coefficients

Wavelets module

- `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.
- `plotwavelets` – common plotting routine.
- Wavelet filters library.
- Helper functions for building FB trees.

Wavelets module – highlights

- Arbitrary number of filters in the basic filterbank – framelets, etc.
- Arbitrary filter trees – DT-$\mathbb{C}$WT
- `fwt2filterbank`, `wfbt2filterbank` – tree filterbank conversion routines using multirate identity.



Filterbanks `(u)filterbank/ifilterbank`

Common routines for FIR, frequency defined and band-limited filters.

$$c_m(n) = \sum_{l=0}^{L-1} f(l)\, g_m(a_m n - l)\,, \tag{1}$$

where $L = k \cdot \mathrm{lcm}(a_m)$, $k \in \mathbb{Z}^+$, $f \in \mathbb{C}^L$ and $a_m n - l$ is computed modulo L.

$$\hat{f}(l) = \sum_{m=0}^{M-1} \sum_{n=0}^{L/a_m - 1} c_m(n)\, \tilde{g}_m(l - a_m n)\,, \tag{2}$$

Filter generating routines:

- `firfilter` – struct, main fields `.h`, `.offset`
- `blfilter` – struct, main fields `.H`, `.foff`

Effective implementation in C.

Filterbanks `(u)filterbank/ifilterbank`

Two purposes:

- A computational routine.
- Filterbank itself as a Frame.

- `filterbankdual`, `filterbankbounds` – dual filterbanks and frame bounds for uniform and painless filterbanks.
- `nonu2ufilterbank` – nonuniform to uniform filterbank transform. Each filter g_m is replaced by $p = \mathrm{lcm}(a_m)/a_m$ delayed versions of itself $z^{-ka_m} G_m(z)$ for $k = 0, \ldots, p-1$

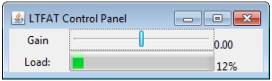
Block-processing framework

A simple framework for a real-time audio processing directly from Matlab/Octave.

```
block('playrec');

p = blockpanel({'GdB','Gain',-20,20,0,21});

while p.flag
   gain = blockpanelget(p,'GdB');
   f = blockread(1024);
   blockplay(f*10^(gain/20));
end
p.close();
```


LTFAT Control Panel

Gain 0.00

Load: 12%

Block-processing framework internals

Based on:
Portaudio (`http://www.portaudio.com`) and
Playrec (`http://www.playrec.co.uk`).

Main features:

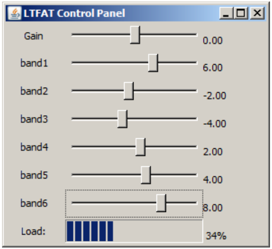
- Interfaces to JACK, ASIO, etc., channel patching.
- No additional toolbox dependency.

Limitations:

- At 44,1 kHz, block sizes $\sim$1000 samples $\Longrightarrow$ latency$\sim$23ms.
- Inherent latency issues from Portaudio.

Block-processing GUI

Configurable control panel



Gain 0.00

band1 6.00

band2 -2.00

band3 -4.00

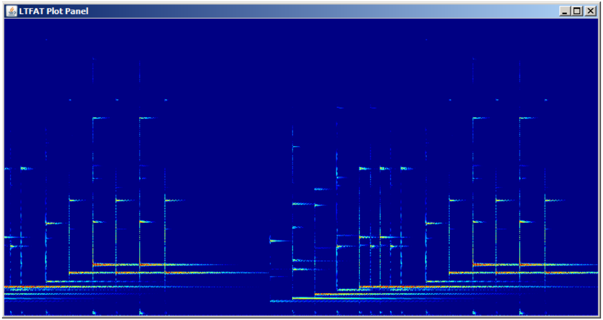
band4 2.00

band5 4.00

band6 8.00

Load: 34%

Real-Time visualization



JAVA based, independent of Matlab GUI framework.

Block-processing with Frames

Basic idea: Analyze (and synthetize) a block stream by any transform available in the Frames framework.

Two issues:

- Speed – backend in C, precomputing using `blockframeaccel`
- Block artifacts
  - Slicing window
  - Overlap-save/ovelap-add

Slicing window

Half-length block overlapping and weighing by a slicing window to reduce time aliasing.

Advantages:

- Works for any transform.
- Delay depends on the block length and is independent of the transform.
- Slicing windows need not add up to 1 – dual slicing window.

Disadvantages:

- Coefficients reflects the shape of the slicing window.
- The blocking artifact can still be perceived.

Overlap-save/ovelap-add

Employs overlap-save method for the analysis and overlap-add method for the synthesis.

Advantages:

- Coefficients can be processed or visualized directly.
- Completely avoids the blocking artifact.

Disadvantages:

- Requires FIR filters/windows.
- Increased processing delay roughly equal to the longest filter/window length.

Live demo

Outlook

- Releasing LTFAT 2.0
- Various interfaces to LTFAT or LTFAT backend.
  - S_TOOLS-ST^{x} – acoustic speech and signal processing application developed at ARI.
  - Sonic Visualizer (`http://www.sonicvisualiser.org/`) – open-source audio visualizing and annotating application.
  - Python bindings
- Better GUI for the frame multiplier editor – `mulaclab`.

Thank you for listening.

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

Download, try, learn, share, contribute...