The Linear Time-Frequency Analysis Toolbox

Peter L. Søndergaard

Austrian Academy of Sciences

February 27, 2013

LTFAT is a modern Matlab/Octave 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

The project was started in 2004, and version 1.0 was released in 2011.

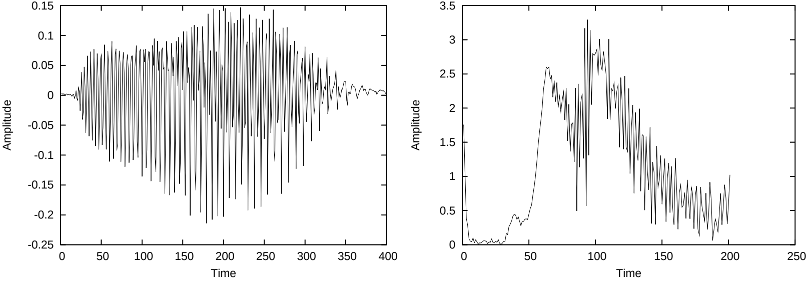
- Basic Fourier analysis and signal processing, FIR windows
- Discrete Gabor transform and its inverse
- Time-frequency bases: Wilson and WMDCT
- Filterbanks and non-stationary Gabor systems
- Methods for constructing perfect reconstruction windows
- Reassignment (sharpening) and instantaneous frequency estimation
- Non-linear analysis and synthesis methods
- Auditory scales and range compression standards

The Discrete Fourier Transform `dft(f)`

$$c(k) = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} f(l) e^{-2\pi i l k}, \quad k = 0, \ldots, L-1$$

- Analyses a signal $f \in \mathbb{C}^L$ in L distinct frequencies (it is an orthonormal basis for $\mathbb{C}^L$)
- Can be computed by a fast $4L\log_2(L)$ algorithm, the FFT
- Output is complex valued even for real-valued input, but you can choose to keep only half the coefficients, or use a discrete cosine/sine transform instead (`dcti`, `dctii`, `dctiii`, `dctiv` or `dsti`, `dstii`, `dstiii`, `dstiv`)

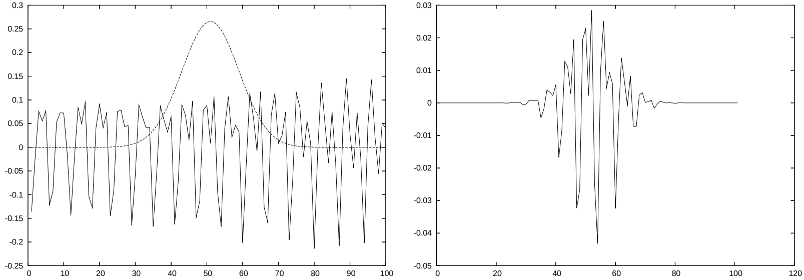
Signal processing



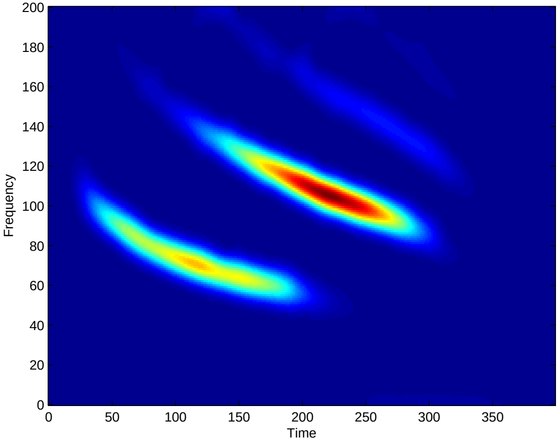
Left plot shows the time representation of the sonar signal of a bat, $f(t)$.
Right plot shows the positive frequency representation, $\hat{f}(\omega)$.

The Short-time Fourier transform

$$\left(STFT_g f\right)(m,n) = \sum_{l=0}^{L-1} f(l)\,\overline{g(l-n)}\,e^{-2\pi i m l/L}, \quad m,n = 0,\ldots,L-1$$



Left plot shows a part of the bat signal overlaid with a Gaussian window.
Right plot shows their pointwise product.



The absolute value squared of the short time Fourier transform is called the spectrogram, `sgram`.

Subsampling the STFT

The Short time Fourier transform is highly redundant:

- Input is in $\mathbb{C}^L$, output is in $\mathbb{C}^{L\times L}$.

Simple solution, we subsample the STFT in a regular fashion:

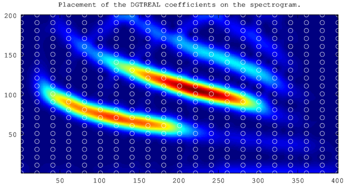
- The Discrete Gabor Transform `dgt(f,g,a,M)`:

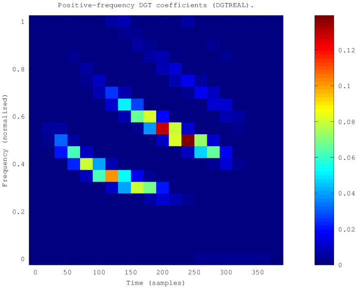
$$\left(V_g f\right)(m,n) = \sum_{l=0}^{L-1} f\left(l\right)\overline{g\left(l-na\right)}e^{-2\pi i m l/M},$$

  where $m = 0,\ldots,M-1$ and $n = 0,\ldots,N-1$ with $L = Na = Mb$ for $N, b \in \mathbb{N}$.

- The output is in $\mathbb{C}^{M\times N}$
- We define the redundancy as the ratio of the number of output to input coefficients:

$$red = \frac{MN}{L} = \frac{M}{a}$$





The points marked by dots on the figure on the left show the placement of the coefficients on the right.

The DGT is not a basis, how can we deal with a redundancy larger than 1?

- A family of vectors f_j is a frame for a Hilbert space $\mathcal{H}$ iff there exists $0 < A \leq B < \infty$ such that

$$A \|f\|^2 \leq \sum_j |\langle f, f_j \rangle|^2 \leq B \|f\|^2, \quad \forall f \in \mathcal{H}.$$

- Upper frame bounds guarantees stability
- Lower framer bounds guarantees invertibility
- The frame operator is given by

$$Sf = \sum_j \langle f, f_j \rangle f_j.$$

- The canonical dual frame (pseudo-inverse) is given by

$$f_j^d = S^{-1} f_j$$

- The canonical tight frame (polar decomposition) is given by

$$f_j^t = S^{-1/2} f_j$$

We can get perfect reconstruction using either construction:

$$f = \sum_j \langle f, f_j \rangle f_j^d$$

$$f = \sum_j \left\langle f, f_j^t \right\rangle f_j^t$$

- Gabor bases are not a good idea: Choose 2:
  - Basis
  - Good localization in both time and frequency
  - A low frame bound ratio $\frac{B}{A}$ (the condition number)
- In the continuous case: The Balian-Low theorem (1981 & 1985). Assume that g creates a Gabor basis for the real line: Then

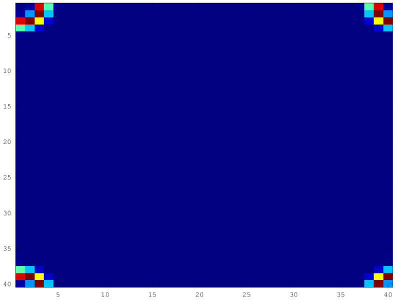
$$\int_{\mathbb{R}} x^2 \left|g\left(x\right)\right|^2 dx = \infty \quad \text{or} \quad \int_{\mathbb{R}} \omega^2 \left|\hat{g}\left(\omega\right)\right|^2 d\omega = \infty$$

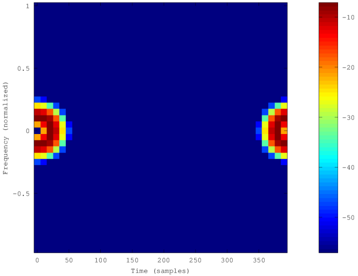
A Gabor frame is an exceptionally good construction, because the frame operator (and its inverse) commutes with time-frequency shifts:

$$g_{m,n}^{d} = S^{-1} g_{m,n} = \left(S^{-1} g\right)_{m,n} = \left(g^{d}\right)_{m,n}$$

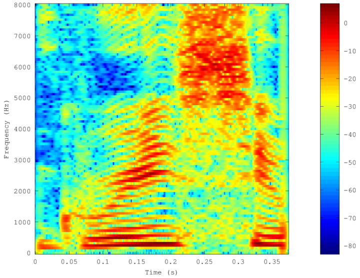
- The canonical dual and tight frames are again Gabor frames
- Therefore, you only need to apply the inverse frame operator once to the window to get them: `gabdual` and `gabtight`.

DGT example



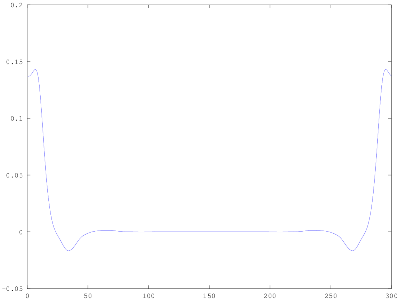



```
a=10; M=40; L=a*M;
h=pherm(L,4); % 4th order Hermite function.
c=dgt(h,'gauss',a,M);
figure(1); imagesc(abs(c).^2);
figure(2); plotdgt(c,a,'dynrange',50);
```

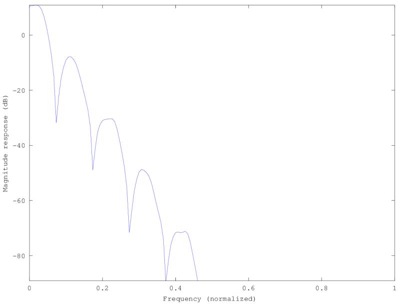


- Most signals occurring in applied sciences are real-valued
- No need to mess with the negative frequencies.
- Analysis: `dgtreal`. Synthesis: `idgtreal`

```
f=greasy; fs=16000;
a=10; M=200;
c=dgtreal(f,{'hann',0.02*fs'},a,M);
plotdgtreal(c,a,M,fs,90);
```



Dual window in time domain



Dual window in freq. domain

- `gabdual`: Compute the window of the canonical dual Gabor frame
- `gabtight`: Compute the window of the canonical tight Gabor frame
- When used together with `dgt`/`idgt` or `dgtreal`/`idgtreal` these windows ensure perfect reconstruction

Is the Balian-Low theorem end of the search for a time-frequency basis with a good resolution?

- Wilson bases / MDCT. A Wilson basis has a linear frequency scale, and is constructed from a Gabor frame with redundancy 2. Discovered by 4 different research groups around 1986 - 1989.
- Wavelets. Wavelets have a logarithmic frequency scale. Morlet 1970s, mathematicians 1980s and onwards, Multiresolution Analysis, S. Mallat 1988/89

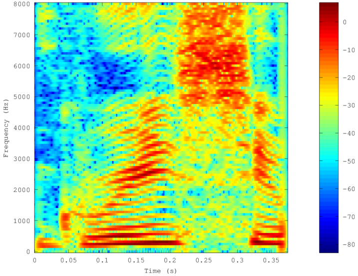
The *Windowed Modified Discrete Cosine Transform*:

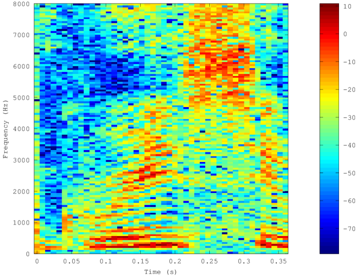
$$c\left(m,n\right) = \sqrt{2}\sum_{l=0}^{L-1} f(l)\cos\left(\frac{\pi}{M}\left(m+\frac{1}{2}\right)l+\frac{\pi}{4}\right)g(l-nM),\; m+n \text{ even}$$

$$c\left(m,n\right) = \sqrt{2}\sum_{l=0}^{L-1} f(l)\sin\left(\frac{\pi}{M}\left(m+\frac{1}{2}\right)l+\frac{\pi}{4}\right)g(l-nM),\; m+n \text{ odd}$$

Methods implemented:

- Transform and inverse: `wmdct` and `iwmdct`
- Riesz dual and orthonormal window: `wildual` and `wilorth`
- Frame bounds: `wilframebounds`
- Wilson transform and inverse: `dwilt` and `idwilt`





`dgtreal` (redundancy=20)   `wmdct` (redundancy=1)

```
fs=16000; % Sampling rate
c=wmdct(greasy,{'hann',0.02*fs'},128);
plotwmdct(c,fs,90);
```

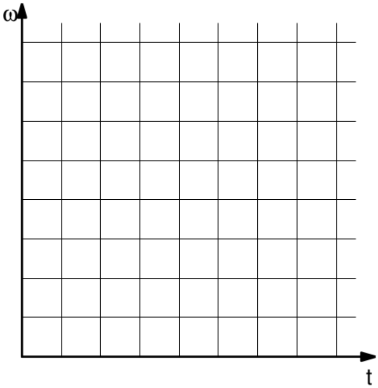
The continuous wavelet transform is defined by

$$W_{\psi}(a,b) = \frac{1}{\sqrt{a}} \int_{\mathbb{R}} f(t)\,\overline{\psi}\left(\frac{t-b}{a}\right) dt, \quad a > 0, b \in \mathbb{R}$$

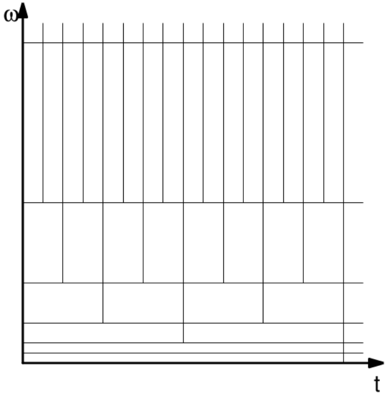
The discrete wavelet transform `fwt` is computed by

$$\begin{aligned} c_0(n) &= f(n) \\ c_{j+1}(n) &= \sum_k c_j(k)\, g(2n-k) \\ d_{j+1}(n) &= \sum_k c_j(k)\, h(2n-k), \end{aligned}$$

where g and h is a pair of quadrature mirror filters (QMF) that splits a signal into its low (g) and high (h) frequency components.



Gabor

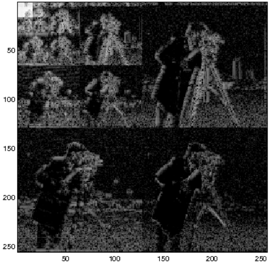


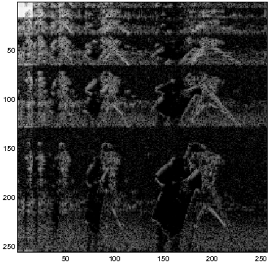
Wavelet

The current constructions

- Fast Wavelet Transform `fwt` : Always split the low frequencies
- Undecimated fwt `ufwt`: No subsampling
- Wavelet filterbank tree `wfbt` : Selectively split low or high frequencies
- Wavelet packet transform `wpfbt`: Split and keep all splittings
- Best basis search `wpbest`: Search for the optimal representation in the wavelet packet

FWT2 example





Standard

Tensor

```
c = fwt2(cameraman,{'db',8},4);
imagesc(dynlimit(20*log10(abs(c)),70));
axis('image'); colormap(gray);
```

Instantaneous frequency

The local instantaneous frequency of a signal can be computed in several ways:

- From the phase (Flanagan 1966)

$$IF\left(x,\omega\right)=\frac{\partial}{\partial x}\angle\left(V_{g}f\right)\left(x,\omega\right)$$

- From the full STFT (Auger 1995)

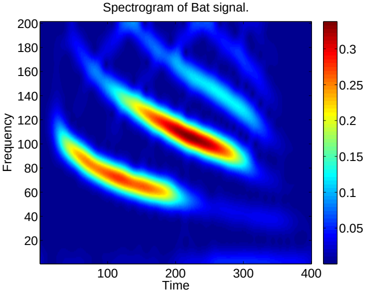
$$IF\left(x,\omega\right)=-\Im\left(\frac{V_{g'}f\left(x,\omega\right)}{V_{g}f\left(x,\omega\right)}\right)$$

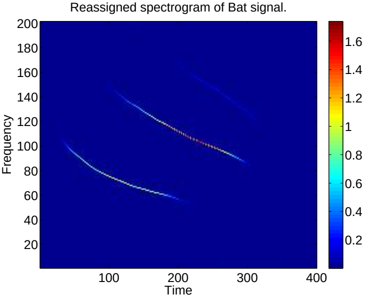
If a signal is well approximated by a sine function close to the point (x,ω) in the TF-plane, the frequency of this sine function is

$$\omega+IF\left(x,\omega\right).$$

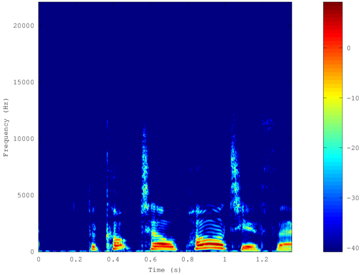
The reassignment process moves a contribution to the spectrogram from the point (x, ω) to the point

$$(x + IT(x, \omega), \omega + IF(x, \omega))$$

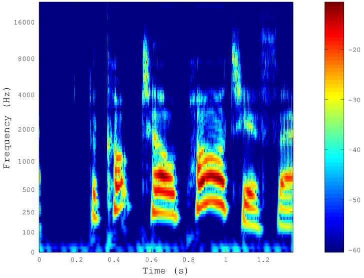




The reassignment process improves the precision of the spectrogram, but the resolution must still obey the uncertainty principle.



Spectrogram



Auditory filterbank

$$c(m,n) = \sum_{l=0}^{L-1} f(l)\,\overline{g_m}(a_m n - l)$$

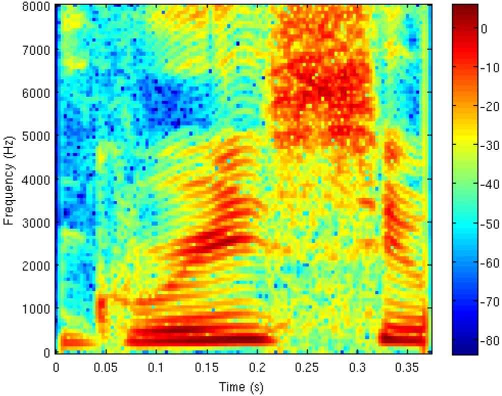
- Sometimes the fixed frequency resolution of the dgt is not desirable
- filterbanks: Each frequency channel is controlled by its own filter g_m

- `filterbank`: Each channel is a subsampling of a convolution with a varying sampling distance a_m
- `ufilterbank`: Uniform filterbank. a has the same value for all channels.
- general filterbanks are very flexible, because the subsampling rate can be adapted to the window
- uniform filterbanks are easily invertible, the canonical dual frame is again a uniform filterbank

- A *frame object* in LTFAT is a struct defining a frame
- The struct is created by `frame`
- The struct is generally read-only
- Simple code for generating canonical dual frames:
  `[Fa,Fs]=framepair('dgt','gauss','dual',40,60);`

The information given to `frame` is not usually not enough to completely determine the frame. The length of the input signal is also needed, allowing the frame definition to be reused for various input signals. Basic properties like redundancy and frame bounds can be established without the length being determined.

- `frame` : Create frame
- `frana` : Frame analysis operator
- `frsyn` : Frame synthesis operator
- `frameaccel` : Speed up frame application
- `plotframe` : Plot frame coefficients




```
F=frame('dgtreal','gauss',50,200);
c=frana(F,greasy);
plotframe(F,c,16000,90);
```

Information about the frame

- `framered` : Redundancy of the frame
- `framebounds` : Frame bounds
- `frameupperbound` : (future) Approx. upper frame bound
- `framemat` : Matrix representation of the synthesis operator
- `framelength` : Length of frame to expand a given signal
- `framelengthcoef` : Length of frame given a set of coefficients

Currently supported frames

General / special frames

- gen : general frame specified by a matrix
- identity : The canonical orthonormal basis

Gabor like frames:

- dgt : Gabor frame
- dgtreal : Gabor frame for real valued signals
- dwilt : Wilson basis
- wmdct : Windowed modified cosine transform

Pure frequency bases

- `dft`
- `dcti, dctii, dctiii, dctiv`
- `dsti, dstii, dstiii, dstiv`

Wavelets

- fwt : Fast Wavelet Transform
- ufwt : Undecimated FWT
- `wfbt` : Wavelet filterbank tree
- `wpfbt` : Wavelet packet transform
- cqt : Constant Q transform

Filterbanks

- filterbank : General filterbank
- ufilterbank : Uniform filterbank
- filterbankreal : Positive-frequency filterbank intended for real-valued signals
- ufilterbankreal : Uniform version of above

Non-stationary Gabor systems

- nsdgt : Non-stationary Gabor system
- unsdgt : Uniform non-stationary Gabor system
- nsdgtreal : Non-stationary Gabor system for real-valued signals
- unsdgtreal : Uniform non-stationary Gabor system for real-valued signals

Sometimes the canonical dual of a frame is not a frame of the same type (i.e. non-uniform filterbanks). In these cases, the following methods can be useful

- `franaiter` : Inverse of the synthesis operator computed using a conjugate gradient (PCG) method
- `frsyniter` : Inverse of the analysis operator computed using a conjugate gradient (PCG) method

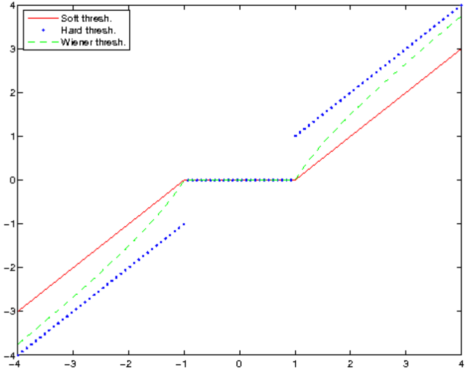
.

Reconstruction from absolute values

In some applications, only the absolute value of the frame coefficients are of interest, and a common problem is to find the signal having absolute value of its frame coefficients closest to a given target

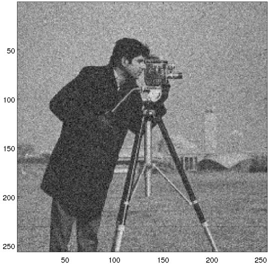
- `frsynabs` : Synthesis from absolute value of frame coefficients

The method uses the Griffin-Lim algorithm, which is an iterative algorithm using a succession of projections of the target onto the reproducing kernel space, each time modifying the phase.



- `thresh` : Thresholding by value
- `largestn` : Keep the N largest coefficients
- `largestr` : Keep the ratio r of the largest coefficients

Denoising by soft thresholding



Noisy image

Soft thresholding

```
c=fwt2(fnoisy,{'db',5},6);
cthresh=largestr(c,0.1,'soft');
fthresh=ifwt2(cthresh,{'db',5},6);
```

- `franalasso` solves the LASSO (or basis pursuit denoising) regression problem for a general frame F: find the coefficients c that minimize

$$\frac{1}{2}\left\|Fc - f\right\|_2 + \lambda\left\|c\right\|_1,$$

where f is the input signal and λ is a penalization coefficient.

- `franagrouplasso` solves the group LASSO regression problem in the time-frequency domain: minimize a functional of the synthesis coefficients defined as the sum of half the l^2 norm of the approximation error and the mixed l^1 / l^2 norm of the coefficient sequence, with a penalization coefficient λ.

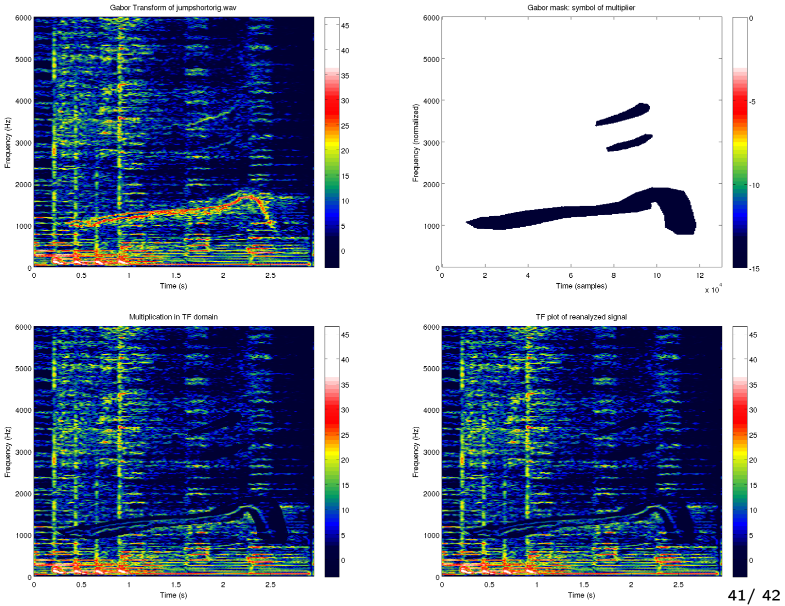
A multiplier is an operator given by

$$h = \sum_{k=0}^{K-1} m(k) \langle f, f_k^a \rangle f_k^s,$$

where m is the symbol of the multiplier and $\{f^a\}$ and $\{f^s\}$ are the analysis and synthesis frames.

- Apply frame multiplier: `framemul`
- Apply the adjoint of a frame multiplier: `framemuladj`
- Apply the inverse of a frame multiplier: `iframemul`
- Best approx. by frame multiplier: `framemulappr`
- Eigenpairs of a frame multiplier: `framemuleigs`

Gabor multiplier example



- Version 2.0 expected around the end of the year:
  - Completed the basic wavelet package
  - Wigner distribution, fractional Fourier transform, prolate spheroidal wave functions
  - Inclusion of YAWTB
  - Block processing interface to work with streaming data
- Thank you very much for coming!