Fast and Memory-Efficient Algorithms for Quadratic Time–Frequency Distributions

A collection of M-files to compute time–frequency distributions (TFDs) from the quadratic class [1]. Memory and computational load is limited by controlling the level of over-sampling for the TFD. Oversampling in the TFD is proportional to signal length and bandwidth of the Doppler–lag kernel. Algorithms are optimised to four kernel types: nonseparable, separable, lag-independent, and Doppler-independent kernels.

Also included are algorithms to compute decimated, or sub-sampled, TFDs. Again, these algorithms are specific to the four kernel types but compute approximate TFDs by a process of decimatation.

Requires Matlab or Octave (programming environments). Latest version available at http://otoolej.github.io/code/memeff_TFDs/.

1 Description

First, add paths using the `load_curdir` function:

```
>> load_curdir;
```

There are two sets of TFD algorithms: one set computes oversampled TFDs and the other set computes decimated (sub-sampled or undersampled) TFDs. The first set, for computing oversampled TFDs, has four algorithms for specific kernel types, namely the

- non-separable,
- separable,
- Doppler-independent (DI),
- lag-independent (LI) kernels.

1.1 Oversampled TFDs

The function to generate these oversampled TFDs is `full_tfd.m`. The following examples, using a test signal, illustrates usage:

```
% generate test signal:
N=512;
x=gen_LFM(N,0.1,0.3) + gen_LFM(N,0.4,0.04);

% nonseparable kernel (Choi-Williams kernel):
```

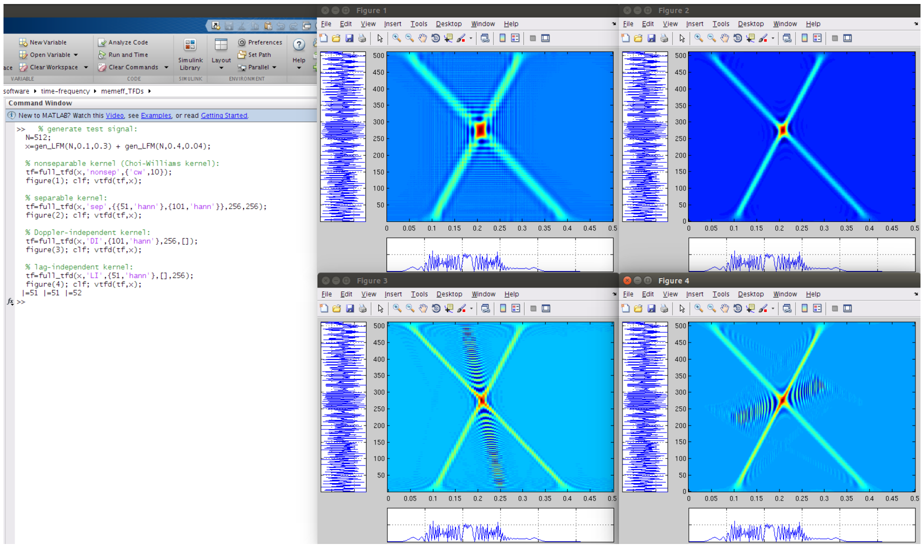

Figure 1: Examples of oversampled TFDs.

```
tf=full_tfd(x,'nonsep',{'cw',10});
figure(1); clf; vtfd(tf,x);

% separable kernel:
tf=full_tfd(x,'sep',{{51,'hann'},{101,'hann'}},256,256);
figure(2); clf; vtfd(tf,x);

% Doppler-independent kernel:
tf=full_tfd(x,'DI',{101,'hann'},256,[]);
figure(3); clf; vtfd(tf,x);

% lag-independent kernel:
tf=full_tfd(x,'LI',{51,'hann'},[],256);
figure(4); clf; vtfd(tf,x);
```

Type `help full_tfd` for more details.

1.2 Undersampled TFDs

Likewise, the algorithms for decimated TFDs are specific to the four kernel types. The function `dec_tfd` computes the decimated TFDs, as the following examples show:

```
N=1024; Ntime=64; Nfreq=128;
a=2; b=2;
ni=[100:2:900]; ki=[150:2:850];
```

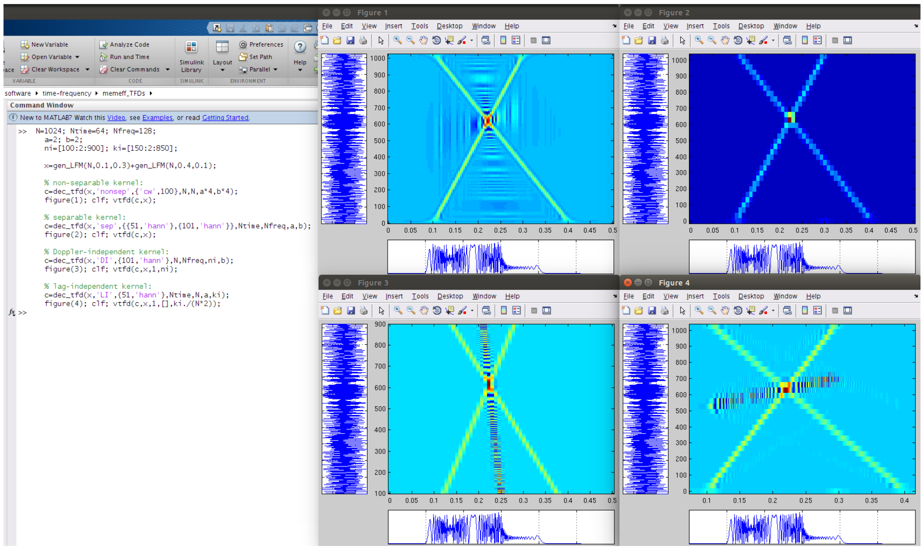

Figure 2: Examples of decimated TFDs.

```

x=gen_LFM(N,0.1,0.3) + gen_LFM(N,0.4,0.1);

% non-separable kernel:
c=dec_tfd(x,'nonsep',{'cw',100},N,N,a*4,b*4);
figure(1); clf; vtfd(c,x);

% separable kernel:
c=dec_tfd(x,'sep',{{51,'hann'},{101,'hann'}},Ntime,Nfreq,
    a,b);
figure(2); clf; vtfd(c,x);

% Doppler-independent kernel:
c=dec_tfd(x,'DI',{101,'hann'},N,Nfreq,ni,b);
figure(3); clf; vtfd(c,x,1,ni);

% lag-independent kernel:
c=dec_tfd(x,'LI',{51,'hann'},Ntime,N,a,ki);
figure(4); clf; vtfd(c,x,1,[],ki./(N*2));
```

Type `help dec_tfd` for more details on this function.

name	type	description
common	dir.	files to generate kernel functions
decimated_TFDs	dir.	files to generate decimated TFD for the 4 kernel types
full_TFDs	dir.	files to generate oversampled TFD for the 4 kernel types
utils	dir.	miscellaneous files
dec_tfd.m	file	computes decimated TFDs
full_tfd.m	file	computes oversampled TFDs
LICENCE.md	file	licence file
README.md	file	README file (markdown format)
README.pdf	file	this README file
load_curdir.m	file	adds paths for Matlab/Octave

Table 1: Files and directories (dir.)

2 Files

All Matlab files (.m files) have a description and an example in the header. To read this header, type `help <filename.m>` in Matlab. Directory structure is in Table 1.

3 Requirements

Either Matlab (R2012 or newer, Mathworks website) or Octave (v3.6 or newer, Octave website, with the 'octave-signal' add-on package).

4 Test Computer

- hardware: Intel(R) Xeon(R) CPU E5-1603 0 @ 2.80GHz; 8GB memory.
- operating system: Ubuntu GNU/Linux x86_64 distribution (Trusty Tahr, 14.04), with Linux kernel 3.13.0-27-generic
- software: Octave 3.8.1 (using Gnuplot 4.6 patchlevel 4), with 'octave-signal' toolbox and Matlab (R2009b, R2012a, and R2013a)

5 References

1. J.M. O' Toole and B. Boashash, "Memory Efficient Algorithms for Quadratic TFDs", Chapter 6.6; in *Time–Frequency Signal Processing and Analysis: A Comprenhensive Reference*, Second Edition, Academic Press, pp. 374–385, 2016 (ISBN: 9780123984999).

2. J.M. O' Toole and B. Boashash, "Fast and memory-efficient algorithms for computing quadratic time–frequency distributions", *Applied and Computational Harmonic Analysis*, vol. 35, no. 2, pp. 350–358, 2013.

3. J.M. O Toole, M. Mesbah, and B. Boashash, “Improved discrete definition of quadratic time–frequency distributions,” *IEEE Transactions on Signal Processing*, vol. 58, Feb. 2010, pp. 906-911.

4. J.M. O’ Toole, M. Mesbah, and B. Boashash, “A New Discrete Analytic Signal for Reducing Aliasing in the Discrete Wigner-Ville Distribution”, *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5427-5434, Nov. 2008.

5. J.M. O Toole, M. Mesbah, and B. Boashash, “Algorithms for discrete quadratic time–frequency distributions,” *WSEAS Transactions on Signal Processing*, vol. 4, May. 2008, pp. 320-329.

6 Contact

Name: John M. O’ Toole

Address: Neonatal Brain Research Group,
Irish Centre for Fetal and Neonatal Translational Research (INFANT),
Department of Paediatrics and Child Health,
University College Cork,
Western Gateway Building, Room 4.01L,
Cork, Ireland

email: j.otoole@ieee.org

web: http://otoolej.github.io/

7 Licence