WWW.KFS.OEAW.AC.AT

ÖAW ÖSTERREICHISCHE AKADEMIE DER WISSENSCHAFTEN

The Phase Retrieval Toolbox

Zdeněk Průša

ISF

PHASERET is a Matlab/GNU Octave toolbox for phase (signal) reconstruction from the short-time Fourier transform (STFT) magnitude. The toolbox provides an implementation of various, conceptually different algorithms including real-time capable algorithms which are also implemented in real-time audio demos running directly in Matlab/GNU Octave. The toolbox depents on the Large Time-Frequency Analysis Toolbox – LTFAT (`http://ltfat.github.io`), it is well-documented, open-source and it is available under the GPL3 license at

`http://ltfat.github.io/PHASERET`

The Problem

The discrete STFT (also known as the discrete Gabor transform – DGT) is defined as

$$\mathbf{c}(m+nM) = \sum_{l=0}^{L-1} \mathbf{f}(l+na)\mathbf{g}(l)\mathrm{e}^{-\mathrm{i}2\pi ml/M}$$

where $n = 0,\ldots,L/a-1$, $m = 0,\ldots,M-1$ and where

- $\mathbf{c} \in \mathbb{C}^{MN}$ is the DGT coefficient vector,
- $\mathbf{f} \in \mathbb{R}^{L}$ is the input signal,
- $\mathbf{g} \in \mathbb{R}^{L}$ is the analysis window,
- a is the time step size,
- M is the number of frequency channels (FFT size).

Alternatively using matrix notation (analysis)

$$\mathbf{c} = \mathbf{F}_{\mathbf{g}}^{*}\mathbf{f},$$

where columns of $\mathbf{F}_{\mathbf{g}} \in \mathbb{C}^{L\times MN}$ are in a form of

$$\mathbf{g}_{m+nM}(l) = \mathbf{g}(l-na)\mathrm{e}^{\mathrm{i}2\pi m(l-na)/M}$$

for $l = 0,\ldots,L-1$. The signal can be recovered using

$$\mathbf{f} = \left(\mathbf{F}_{\mathbf{g}}^{*}\right)^{\dagger}\mathbf{c} = \left(\mathbf{F}_{\mathbf{g}}\mathbf{F}_{\mathbf{g}}^{*}\right)^{-1}\mathbf{F}_{\mathbf{g}}\mathbf{c} = \mathbf{F}_{\tilde{\mathbf{g}}}\mathbf{c}$$

where

$$\tilde{\mathbf{g}} = \left(\mathbf{F}_{\mathbf{g}}\mathbf{F}_{\mathbf{g}}^{*}\right)^{-1}\mathbf{g}.$$

The magnitude and phase components of the coefficients are obtained as

$$\mathbf{s}(p) = |\mathbf{c}(p)|, \quad \phi(p) = \arg\left(\mathbf{c}(p)\right),$$

for $p = 0,\ldots,MN-1$, where $\mathbf{s}$ is also referred to as the *consistent* STFT magnitude (spectrogram) i.e. there exists some signal $\mathbf{f}$ with such STFT magnitude.

Phase retrieval algorithms reconstruct phase $\widehat{\phi}$ and signal $\widehat{\mathbf{f}}$ from (possibly inconsistent) magnitude $\mathbf{s}$.

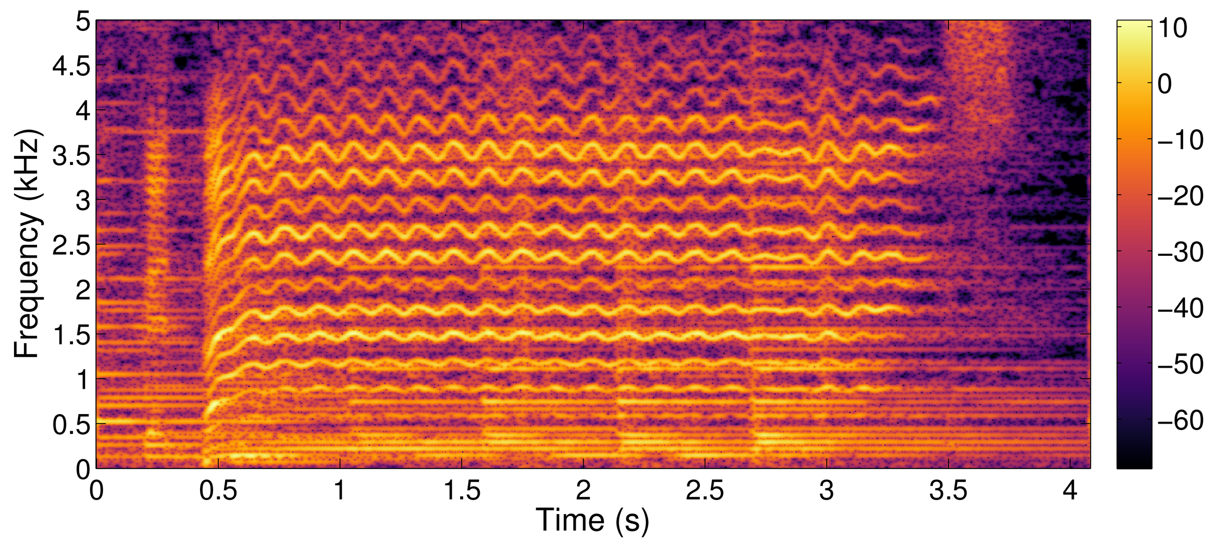


Figure: A spectrogram of a piano and a solo singing voice. An excerpt from the song *Gate 21* performed by *Serj Tankian*.

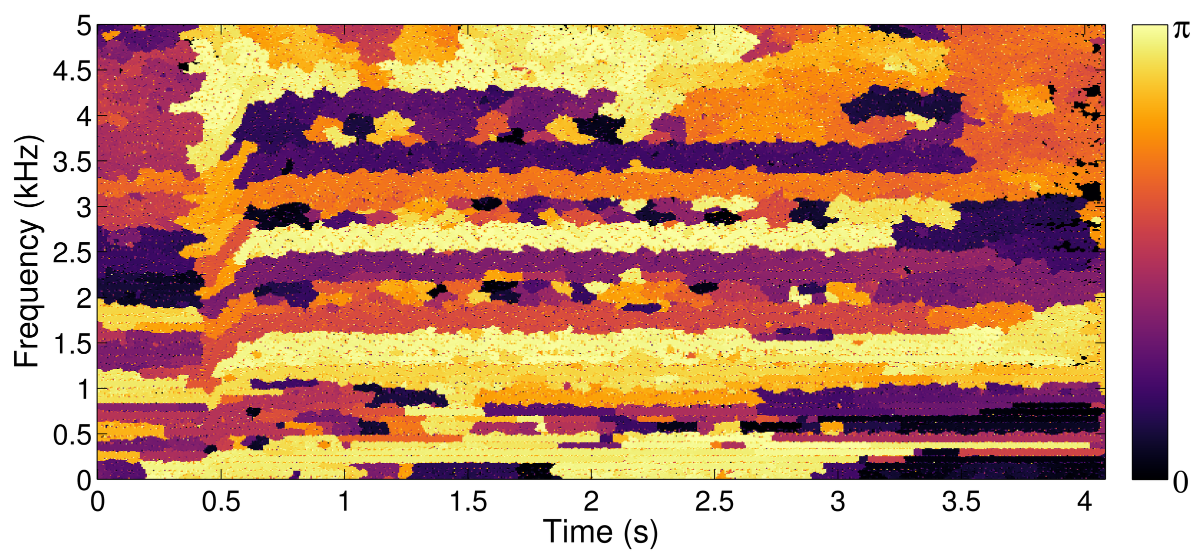


Figure: Phase difference pattern achieved by pghi

$$\min\left(\left|\phi-\widehat{\phi}\right|, 2\pi-\left|\phi-\widehat{\phi}\right|\right)$$

Offline algorithms

Griffin & Lim algorithm (GLA) – **`gla`** [1] proceeds by projecting the coefficients iteratively onto two subsets of $\mathbb{C}^{MN}$ denoted as $\mathcal{C}_1$ and $\mathcal{C}_2$. The set $\mathcal{C}_1$ is identical with the range (column) space of $\mathbf{F}_{\mathbf{g}}^{*}$ i.e.

$$\mathcal{C}_1 = \left\{\mathbf{c} | \mathbf{c} = \mathbf{F}_{\mathbf{g}}^{*}\mathbf{f} \text{ for all } \mathbf{f} \in \mathbb{C}^{L}\right\} \text{ and its orthogonal complement is } \mathcal{C}_1^{\perp} = \left\{\mathbf{c} | \mathbf{F}_{\tilde{\mathbf{g}}}\mathbf{c} = 0\right\}.$$

Any $\mathbf{c} \in \mathbb{C}^{MN}$ can be written as a sum of components in these two spaces as $\mathbf{c} = \mathbf{c}^{\mathcal{C}_1} + \mathbf{c}^{\mathcal{C}_1^{\perp}}$ and the unique orthogonal projection onto $\mathcal{C}_1$ is

$$\mathbf{P}_{\mathcal{C}_1}\mathbf{c} = \mathbf{F}_{\mathbf{g}}^{*}\left(\mathbf{F}_{\mathbf{g}}\mathbf{F}_{\mathbf{g}}^{*}\right)^{-1}\mathbf{F}_{\mathbf{g}}\mathbf{c} = \mathbf{F}_{\mathbf{g}}^{*}\mathbf{F}_{\tilde{\mathbf{g}}}\mathbf{c} = \mathbf{c}^{\mathcal{C}_1},$$

The set $\mathcal{C}_2$ is a set of coefficients with magnitude equal to the target magnitude $\mathbf{s}$

$$\mathcal{C}_2 = \left\{\mathbf{c} | \, |\mathbf{c}(p)| = \mathbf{s}(p) \text{ for all } p\right\}.$$

Since it is not a convex set, the projection is substituted simply by forcing the magnitude of the coefficients to the target magnitude

$$\left(\mathbf{P}_{\mathcal{C}_2}\mathbf{c}\right)(p) = \mathbf{s}(p)\mathbf{c}(p)/\left|\mathbf{c}(p)\right| \text{ for all } p.$$

The composition of the projections then gives i-th iteration of GLA

$$\mathbf{c}_i = \mathbf{P}_{\mathcal{C}_2}\mathbf{P}_{\mathcal{C}_1}\mathbf{c}_{i-1}.$$

given $\mathbf{c}_0(p) = \mathbf{s}(p)\mathrm{e}^{\mathrm{i}\phi_0(p)}$ for all p, where $\mathbf{s}$ is the target magnitude and ϕ_0 is the initial phase. An acceleration technique proposed in [2].

Le Roux's Modifications of GLA (leGLA) – **`legla`** [3] Modification 1) exploiting a structure in the projection $\mathbf{P}_{\mathcal{C}_1}$ and introducing its approximation using a truncated kernel. Modification 2), since this way the projection $\mathbf{P}_{\mathcal{C}_1}$ can be done coefficient-wise and so can be done $\mathbf{P}_{\mathcal{C}_2}$, the authors proposed to do *on-the-fly* updates i.e. to reuse the just updated coefficient for computing others within the same iteration. Modification 3) Introduction of a *modified* phase update (omitting contribution of the coefficient being updated).

Decorsiere's Unconstrained Optimization – **`decolbfgs`** [4] uses the lBFGS algorithm for minimizing a cost function $\mathcal{G}(\mathbf{f}) = \left\|\left|\mathbf{F}_{\mathbf{g}}^{*}\mathbf{f}\right|^{q} - \mathbf{s}^{q}\right\|^{2}$ while exploiting an explicit formula for its gradient.

Phase Gradient Heap Integration (PGHI) – **`pghi`** [5] reconstruction method based on the relationship between the magnitude and the phase gradients. Assuming the Gaussian window is used, the phase gradient can be estimated using the magnitude gradient and the phase is then recovered by employing an adaptive integration procedure. Works well also for non-Gaussian windows.

Online algorithms – Offline Implementation

Single Pass Spectrogram Inversion (SPSI) – **`spsi`** [6] is a phase phase-locked vocoder-like technique. It employs peak picking followed by quadratic interpolation of the magnitude in order to precisely identify the instantaneous frequency. The inst. frequency gives the increment of the peak phase between two frames. The peak phase is also assigned to all coefficients within its region of influence.

Real-Time Phase Gradient Heap Integration (RTPGHI) – **`rtpghi`** [7] is a real-time version of `pghi`. The core of the algorithm was kept, with the exception that the phase is spread only from the previous to the current frame. It is available in one and zero look-ahead frames versions.

Real-Time Iterative Spectrogram Inversion with Look Ahead (RTISI-LA) – **`rtisila`** [8] employs N_{LA} look-ahead frames and proceeds by doing GLA-type iterations on individual frames from the most recent one to the current one and repeats this procedure for a pre-determined number of iterations. An inherent problem is an asymmetry introduced by windowing the partially reconstructed signal. In order to compensate for this problem, the authors proposed to use two additional asymmetric analysis windows to be used with the most recent look-ahead frame and to use the regular analysis window for other frames.

Gnann and Spiertz's RTISI-LA (GSRTISI-LA) – **`gsrtisila`** [9] introduces an alternative way of dealing with the asymmetry of the partially reconstructed signal. It uses $N_{\mathrm{LA}}+1$ additional analysis windows. A combination with RTPGHI proved to give excellent results [10].

Basic usage

Listing 1: Code example of a typical workflow

```
% Load the glockenspiel test signal
% Sampling rate fs = 44100, 262144 samples
[f,fs] = gspi;
a = 256;  % Time step
M = 2048; % Number of frequency channels
g = 'blackman';     % Blackman window
gtilde = {'dual',g}; % Synthesis window
% Compute the DGT coefficients c, Ls=numel(f)
[c,Ls] = dgtreal(f,g,a,M,'timeinv');
% Obtain the magnitude
s = abs(c);
% Reconstruct the phase
chat = phase_rec_func(s,...
% Reconstruct the signal
fhat = idgtreal(chat,gtilde,a,M,Ls,'timeinv');
```

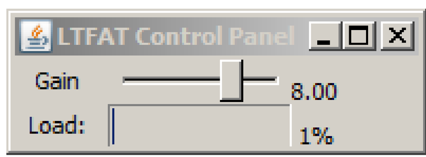
Real-time Implementation

Exploits the block processing framework for real-time audio processing from LTFAT:

Listing 2: Minimal working example

```
% Create the GUI panel
p = blockpanel({'GdB','Gain',-20,20,0,21});
% Initialize the audio device
block('playrec','loadind',p);
% Start audio processing loop
while p.flag
    gain = blockpanelget(p,'GdB');
    f = blockread();
    blockplay(f*10^(gain/20));
end
% Cleanup
blockdone(p);
```

Figure: JAVA control panel



- `demo_blockproc_phaseret` – comparison of RTPGHI, SPSI and RTISI-LA (from [7])
- `demo_blockproc_phaseret2` – comparison of RTPGHI, RTISI-LA, GSRTISI-LA and RTPGHI + GSRTISI-LA (from [10])
- `demo_blockproc_phaseretmix` – comparison of algorithms doing comb-filter free channel mixing (from [7])

Toolbox Organization and Documentation

- Public GitHub repository `https://github.com/ltfat/phaseret`
- C backend library with MEX interfaces `https://github.com/ltfat/libphaseret`
- Web doc. pages generated from the m-file headers using home-brewed system `mat2doc` `https://github.com/ltfat/mat2doc`

References

[1] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 236–243, Apr 1984.

[2] N. Perraudin, P. Balazs, and P. L. Søndergaard, "A fast Griffin-Lim algorithm," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Oct 2013, pp. 1–4.

[3] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in *Proc. 13th Int. Conf. on Digital Audio Effects (DAFx-10)*, Sep. 2010, pp. 397–403.

[4] R. Decorsiere, P. L. Søndergaard, E. N. MacDonald, and T. Dau, "Inversion of auditory spectrograms, traditional spectrograms, and other envelope representations," *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 46–56, Jan 2015.

[5] Z. Průša, P. Balazs, and P. L. Søndergaard, "A Noniterative Method for Reconstruction of Phase from STFT Magnitude," *IEEE/ACM Trans. on Audio, Speech, and Lang. Process.*, vol. 25, no. 5, May 2017.

[6] G. T. Beauregard, M. Harish, and L. Wyse, "Single pass spectrogram inversion," in *Digital Signal Processing (DSP), IEEE International Conference on*, July 2015, pp. 427–431.

[7] Z. Průša and P. L. Søndergaard, "Real-Time Spectrogram Inversion Using Phase Gradient Heap Integration," in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Sep 2016, pp. 17–21.

[8] X. Zhu, G. T. Beauregard, and L. Wyse, "Real-time signal estimation from modified short-time Fourier transform magnitude spectra," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1645–1653, July 2007.

[9] V. Gnann and M. Spiertz, "Improving RTISI phase estimation with energy order and phase unwrapping," in *Proc. 13th Int. Conf. Digital Audio Effects (DAFx-10)*, Sep. 2010, pp. 1–5.

[10] Z. Průša and P. Rajmic, "Toward high-quality real-time signal reconstruction from STFT magnitude," *IEEE Signal Processing Letters*, vol. 24, no. 6, June 2017.

This work was supported by the Austrian Science Fund (FWF): Y 551–N13.

ISF-INSTITUT FÜR SCHALLFORSCHUNG